



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - TE141599
DETEKSI KETIDAKNORMALAN KOPLING PADA MOTOR
INDUKSI MENGGUNAKAN SENSOR ACCELEROMETER
BERBASIS RASPBERRY PI

Rahmadi Ridho Anjiko
NRP 2211 100 142

Dosen Pembimbing
Dr. Muhammad Rivai, ST., MT.
Suwito, ST., MT.

JURUSAN TEKNIK ELEKTRO
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2015



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - TE141599
DETEKSI KETIDAKNORMALAN KOPLING PADA MOTOR
INDUKSI MENGGUNAKAN SENSOR ACCELEROMETER
BERBASIS RASPBERRY PI

Rahmadi Ridho Anjiko
NRP 2211 100 142

Dosen Pembimbing
Dr. Muhammad Rivai, ST., MT.
Suwito, ST., MT.

JURUSAN TEKNIK ELEKTRO
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2015



ITS
Institut
Teknologi
Sepuluh Nopember

FINAL PROJECT - TE 141599

**ABNORMALITIES CLUTCH DETECTION IN INDUCTION
MOTOR USING ACCELEROMETER SENSOR BASED ON
RASPBERRY PI**

Rahmadi Ridho Anjiko
NRP 2211 100 142

Advisor
Dr. Muhammad Rivai, ST., MT.
Suwito, ST., MT.

ELECTRICAL ENGINEERING DEPARTEMENT
Faculty of Industrial Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2015

LEMBAR PENGESAHAN

DETEKSI KETIDAK NORMALAN KOPLING PADA MOTOR INDUKSI MENGGUNAKAN SENSOR ACCELEROMETER BERBASIS RASPBERRY PI

TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik
Pada

Bidang Studi Elektronika
Jurusan Teknik Elektro
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember

Menyetujui:

Dosen Pembimbing I,

Dosen Pembimbing II,

Dr. Muhammad Rivai, ST., MT.
NIP. 1969 04 26 1994 03 1003

Suwito, ST., MT.
NIP. 1981 01 05 2005 01 1004

SURABAYA
JULI, 2015

DETEKSI KETIDAKNORMALAN KOPLING PADA MOTOR INDUKSI MENGGUNAKAN SENSOR ACCELEROMETER BERBASIS RASPBERRY PI

Nama : Rahmadi Ridho Anjiko
NRP : 2211100142
Jurusan : Teknik Elektro
Dosen Pembimbing : 1. Dr. Muhammad Rivai. ST., MT.
2. Suwito, ST., MT.

ABSTRAK

Motor induksi sangat banyak digunakan dalam kehidupan sehari-hari kita. Baik itu digunakan di industri maupun di rumah tangga karena konstruksinya sederhana, pengoperasian mudah dan mempunyai kecepatan yang relative konstan. Pemeliharaan/perawatan mesin sangat dibutuhkan untuk memperpanjang umur pemakaian motor induksi. Tindakan preventif harus didahulukan untuk mencegah kerusakan yang lebih besar pada mesin. Salah satu tindakan preventif yang dapat dilakukan adalah monitoring vibrasi pada mesin.

Kerusakan pada motor induksi yang tidak terdeteksi pada tahap awal dapat menyebabkan kerusakan lebih parah. Kerusakan motor di industri dapat mengakibatkan kerugian yang sangat besar karena proses produksi berhenti. Dalam penelitian ini, digunakan sensor accelerometer untuk melihat pola vibrasi sinyal dengan menggunakan Fast Fourier Transform (FFT) dan Neural Network terhadap jenis-jenis kondisi abnormal. Ada tiga jenis kondisi abnormal yang dibahas dalam penelitian ini yaitu *misalignment*, *unbalance* dan *looseness*. Memonitoring keadaan mesin listrik secara kontinyu sehingga keadaan abnormal pada motor listrik dapat diketahui secara dini.

Sistem dapat mendeteksi jenis-jenis kerusakan secara on-line pada ketidaknormalan kopling motor listrik yaitu *unbalance*, *misalignment* dan *looseness*. Tingkat keberhasilan neural network dalam sistem ini mencapai 80 persen.

Kata Kunci: *Fast fourier transform, Motor induksi, Neural Network, Sensor accelerometer, Vibrasi.*



Halaman ini sengaja dikosongkan

ABNORMALITIES CLUTCH DETECTION IN INDUCTION MOTOR USING ACCELEROMETER SENSOR BASED ON RASPBERRY PI

Name : Rahmadi Ridho Anjiko
NRP : 2211100142
Department : Electrical Engineering
Adivsor : 1. Dr. Muhammad Rivai, ST., MT.
2. Suwito, ST., MT.

ABSTRACT

Induction motors are very widely used in our daily lives. Whether it is used in industry and in households because their construction is simple, easy operation and has a relatively constant speed. Engine maintenance is needed to extend the life of the induction motor. Preventive measures should be prioritized in order to prevent greater damage to the machine. One of the preventive measures that can be done is monitoring vibrations in the engine.

Damage in the induction motor is not detected at an early stage can cause more severe damage. Damage to the motor in the industry can result in huge losses due to the production process stops. In this study, uses accelerometer sensor to see the pattern of vibration signals using Fast Fourier Transform (FFT) and Neural Network on the types of abnormal conditions. There are three types of abnormal conditions that are discussed in this study are misalignment, unbalance and looseness. Monitoring the state of the electric machine continuously so that the abnormal state of the electrical machine can be known at an early stage.

The system can detect the type of damage on the on-line coupling an electric motor abnormalities such as unbalance , misalignment and looseness . The success rate of neural network in the system reached 80 percent.

Keywords: *Fast fourier transform, Induction Motor, Neural Network, Sensor accelerometer, Vibration.*



Halaman ini sengaja dikosongkan

KATA PENGANTAR

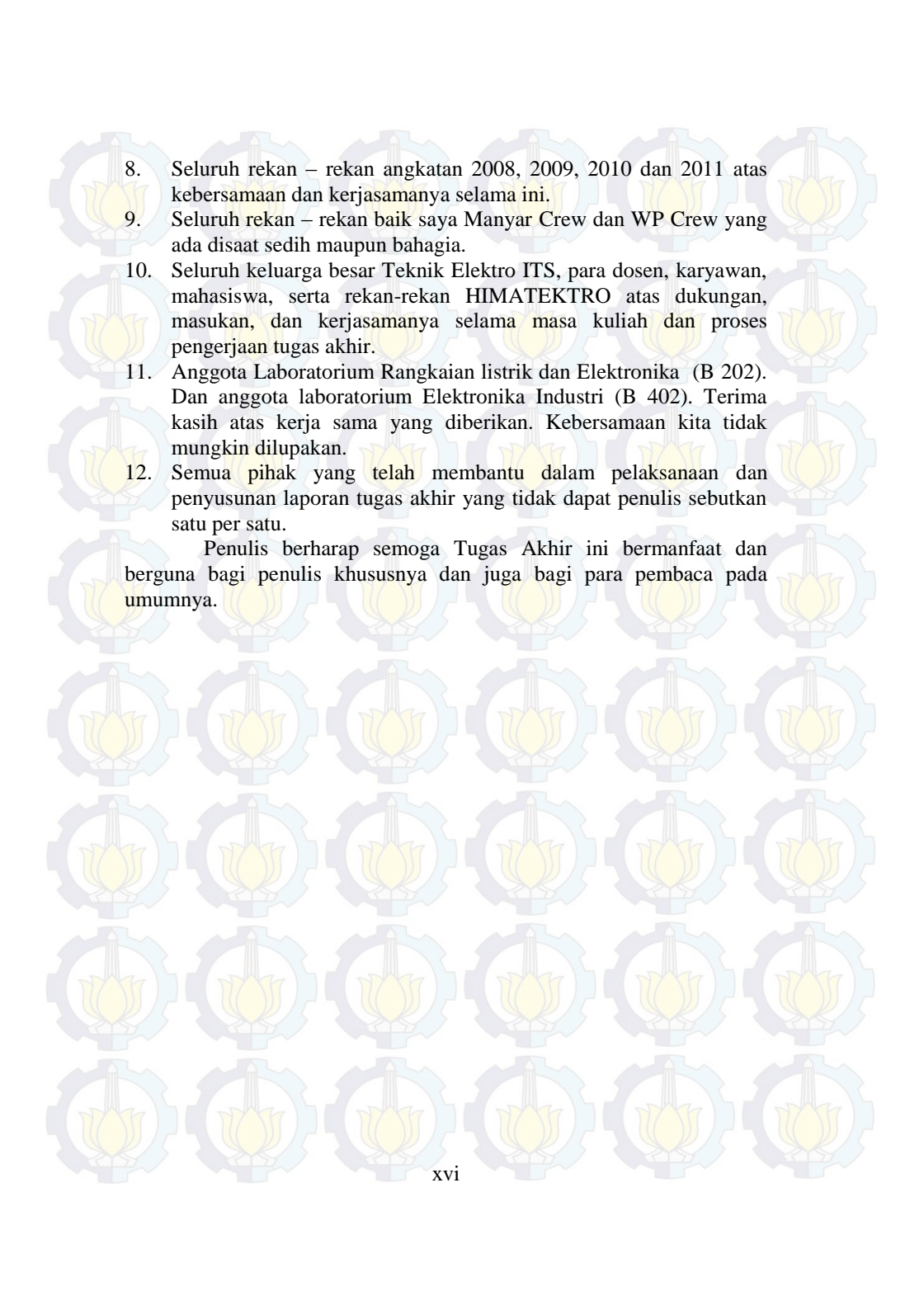
Segala puji syukur bagi Tuhan Yang Maha Esa atas kemudahan, kesehatan, dan kesempatan yang telah dikaruniakan dari awal hingga akhir pengerjaan tugas akhir berjudul:

Deteksi Ketidaknormalan Kopling pada Motor Induksi menggunakan Sensor Accelerometer berbasis Raspberry Pi

Penyusunan Tugas Akhir ini adalah sebagai salah satu persyaratan untuk menyelesaikan jenjang pendidikan S1 pada bidang studi Elektronika di Jurusan Teknik Elektro ITS Surabaya, selain itu juga diharapkan mampu menambah wawasan dalam bidang keahlian elektronika.

Dalam penyusunan Tugas Akhir ini dan selama studi di Institut Teknologi Sepuluh Nopember, penulis mendapatkan bantuan, bimbingan, dan dukungan tak ternilai dari berbagai pihak. Untuk itu penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Bapak Dr. Muhammad Rivai, ST., MT. selaku dosen pembimbing pertama, atas kesediaannya memberikan masukan dan waktu untuk menyelesaikan tugas akhir ini.
2. Bapak Suwito, ST., MT. dosen pembimbing kedua, atas bimbingan, perhatian dan penjelasan serta motivasi yang diberikan.
3. Seluruh Dosen Bidang Studi Elektronika Jurusan Teknik Elektro ITS, atas bimbingan serta ilmu - ilmu perkuliahan yang telah diberikan.
4. Kedua orang tua yang tercinta: Sunaryo dan Siti Samsiyah yang selalu memberikan do'a, nasihat, dukungan serta motivasi yang telah diberikan untuk keberhasilan penulis.
5. Saudara saya yang tercinta: Rizal, Rissa yang membut penulis termotivasi untuk menjadi teladan sejati sebagai adik yang baik.
6. Keluarga Besar E 51 Elektro ITS yang selalu memberi semangat dan selalu membantu memberi motivasi yang sangat luar biasa.
7. Teman – teman seperjuangan :Nafi, Bintang, Kuncoro, Daus, Imam, Paulus, Helmi, dan khususnya Reza dan agung atas waktu, ide – ide, semangat dan dukungan yang diberikan kepada penulis. Terima kasih atas kerja sama yang diberikan. Kebersamaan kita tidak mungkin dilupakan.

- 
8. Seluruh rekan – rekan angkatan 2008, 2009, 2010 dan 2011 atas kebersamaan dan kerjasamanya selama ini.
 9. Seluruh rekan – rekan baik saya Manyar Crew dan WP Crew yang ada disaat sedih maupun bahagia.
 10. Seluruh keluarga besar Teknik Elektro ITS, para dosen, karyawan, mahasiswa, serta rekan-rekan HIMATEKTRO atas dukungan, masukan, dan kerjasamanya selama masa kuliah dan proses pengerjaan tugas akhir.
 11. Anggota Laboratorium Rangkaian listrik dan Elektronika (B 202). Dan anggota laboratorium Elektronika Industri (B 402). Terima kasih atas kerja sama yang diberikan. Kebersamaan kita tidak mungkin dilupakan.
 12. Semua pihak yang telah membantu dalam pelaksanaan dan penyusunan laporan tugas akhir yang tidak dapat penulis sebutkan satu per satu.

Penulis berharap semoga Tugas Akhir ini bermanfaat dan berguna bagi penulis khususnya dan juga bagi para pembaca pada umumnya.

DAFTAR ISI

PERNYATAAN KEASLIAN TUGAS AKHIR	vii
LEMBAR PENGESAHAN	ix
ABSTRAK	xi
ABSTRACT	xiii
KATA PENGANTAR	xv
DAFTAR ISI	xvii
DAFTAR GAMBAR	xxi
DAFTAR TABEL	xxv
BAB 1 PENDAHULUAN	1
1.1. Latar Belakang.....	1
1.2. Perumusan Masalah.....	2
1.3. Batasan Permasalahan	2
1.4. Tujuan Permasalahan.....	3
1.5. Metode Penelitian	3
1.6. Sistematika Penulisan	4
BAB 2 TINJAUAN PUSTAKA	5
2.1. Analisa Vibrasi	5
2.1.1. Analisa Vibrasi dan Noise pada Motor Induksi – Sumber Vibrasi dan Metode Diagnosanya.....	5
2.1.2. Dasar-dasar Monitor Vibrasi Untuk Mendeteksi Kegagalan dan Control Proses	5
2.1.3. Analizer FFT Untuk Diagnosa Kegagalan pada Mesin Berputar Berdasarkan Analisa Vibrasi.....	6
2.2. Motor Induksi	6
2.3. Getaran	7

2.3.1.	Penyebab Vibrasi.....	9
2.3.2.	Accelerometer MEMS (Micro-Electro Mechanical System).....	10
2.3.3.	Accelerometer ADXL345 MEMS.....	12
2.4.	Raspberry Pi.....	13
2.5.	Arduino Uno.....	15
2.6.	Analisa Data Pengukuran Vibrasi	16
2.6.1.	Ketidakeimbangan (Unbalance)	16
2.6.2.	Pompa dan Motor Yang Tidak Selurus (Pump and Motor Misalignment)	17
2.6.3.	Identifikasi Model-model Vibrasi Kondisi <i>Unfault</i> dan <i>Fault</i>	18
2.7.	Fast Fourier Transform.....	19
2.8.	Neural Network (Jaringan Saraf Tiruan)	21
BAB 3 PERANCANGAN SISTEM		25
3.1.	Sistem Identifikasi Vibrasi Saat Melakukan Pengukuran On-line.....	25
3.1.1.	Sensor Accelerometer.....	25
3.1.2.	Raspberry Pi	26
3.1.3.	FFT (Fast Fourier Transform)	27
3.1.4.	Neural Network	30
3.2.	Sistem Pembelajaran Neural Network	33
3.2.1.	Pengambilan Data Frekuensi Untuk ke-4 Pola Vibrasi	33
3.2.2.	Pembelajaran dengan Lazarus	34
3.3.	Metode Pengambilan Data	37
3.3.1.	Model Kondisi Unfault (Kondisi normal)	37
3.3.2.	Model Kondisi Unbalance (Ketidakseimbangan).....	37

3.3.3.	Model Kondisi Misalignment (Tidak Selurus)	39
3.3.4.	Model Kondisi Looseness (Kondisi base kendur)	40
BAB 4 HASIL DAN ANALISA DATA		41
4.1.	Pola-pola Spektrum Vibrasi pada Prototipe Motor Induksi ..	41
4.1.1.	Spektrum Kondisi <i>Unfault</i> (Normal)	43
4.1.2.	Spektrum Kondisi <i>Unbalance</i> (Ketidakseimbangan).....	43
4.1.3.	Spektrum Kondisi <i>Misalignment</i> (Tidak selurus)	45
4.1.4.	Spektrum Kondisi <i>Looseness</i> (Kendur/longgar)	46
4.2.	Klasifikasi Menggunakan Neural Network	47
BAB 5 PENUTUP		53
5.1.	Kesimpulan.....	53
5.2.	Saran.....	53
DAFTAR PUSATAKA		55
LAMPIRAN		57
RIWAYAT HIDUP		77



Halaman ini sengaja dikosongkan

DAFTAR TABEL

Tabel 3.1 Tabel hubungan pola vibrasi dengan NN 31

Tabel 4.1 Hasil Pengujian NN 52



Halaman ini sengaja dikosongkan

DAFTAR GAMBAR

Gambar 1.1 Motor Induksi	1
Gambar 2.1 Motor Induksi	6
Gambar 2.2 Gerak bebas Spring yang berosilasi	8
Gambar 2.3 Struktur MEMS	11
Gambar 2.4 Sensor ADXL345 dan blok diagramnya	13
Gambar 2.5 Raspberry Pi model B+	14
Gambar 2.6 Spesifikasi Raspberry Pi B+	14
Gambar 2.7 Arduino Uno Modul	15
Gambar 2.8 Lokasi pengukuran vibrasi dan pembacaan FFT yang menunjukan puncak vibrasi karena unbalance.....	17
Gambar 2.9 Misalignment angular	17
Gambar 2.10 Misalignment Paralel.....	18
Gambar 2.11 Spektrum vibrasi untuk kondisi unfault dan beberapa kondisi fault	19
Gambar 2.12 Spektrum frekuensi kondisi kerusakan bearing pada outer ball pass (OBPF).....	19
Gambar 2.13 Sinyal fungsi waktu hasil pengukuran vibrasi	20
Gambar 2.14 Sinyal sinus dalam domain waktu dan frekuensi.....	21
Gambar 2.15 Jaringan saraf dengan 3 lapisan	22
Gambar 2.16 Arsitektur Backpropagation	23
Gambar 3.1 Blok diagram sistem vibrasi pengukuran on-line	25
Gambar 3.2 GPIO Raspberry dan Arduino	26
Gambar 3.3 Sinyal domain waktu keluaran accelerometer	27
Gambar 3.4 Flow chart proses FFT dengan sample data vibrasi 1000 data	29

Gambar 3.5 Grafik sinyal dalam domain frekuensi	30
Gambar 3.6 Grafik domain frekuensi	30
Gambar 3.7 Arsitektur neural network dalam penelitian ini	31
Gambar 3.8 Flow chart proses neural network untuk identifikasi pola vibrasi	33
Gambar 3.9 Blok diagram proses pembelajaran NN dengan Lazarus	34
Gambar 3.10 Flow chart proses pembelajaran NN dengan Lazarus ...	35
Gambar 3.11 Hubungan Kopling pompa dan motor saat Unfault.....	37
Gambar 3.12 Posisi baut pada prototype base pompa dan motor	38
Gambar 3.13 Base motor dan pompa saat Unbalance kondisi 1	38
Gambar 3.14 Base motor dan pompa saat Unbalance kondisi 2	39
Gambar 3.15 Base motor dan pompa saat Unbalance kondisi 3	39
Gambar 3.16 Base motor dan pompa saat Unbalance kondisi 4	39
Gambar 3.17 Model kondisi Misalignment (Tidak Selurus)	40
Gambar 4.1 Foto rangkaian prototipe saat pengambilan data	41
Gambar 4.2 Tampilan sinyal pembuktian FFT	42
Gambar 4.3 Domain waktu kondisi unfault	42
Gambar 4.4 Spektrum frekuensi kondisi normal	43
Gambar 4.5 Spektrum frekuensi kondisi normal 2	43
Gambar 4.6 Spektrum frekuensi unbalance kondisi 1	44
Gambar 4.7 Spektrum frekuensi unbalance kondisi 2	44
Gambar 4.8 Spektrum frekuensi unbalance kondisi 3	44
Gambar 4.9 Spektrum frekuensi unbalance kondisi 4	44
Gambar 4.10 Spektrum frekuensi misalignment 1	46
Gambar 4.11 Spektrum frekuensi misalignment 2	46

Gambar 4.12 Spektrum frekuensi looseness kondisi 1	47
Gambar 4.13 Spektrum frekuensi looseness kondisi 2	47
Gambar 4.14 Proses pembelajaran NN di laptop	47
Gambar 4.15 Hasil neural network kondisi unfault	48
Gambar 4.16 Spektrum vibrasi kondisi unfault	48
Gambar 4.17 Hasil neural network kondisi unbalance	49
Gambar 4.18 Spektrum vibrasi kondisi unbalance	49
Gambar 4.19 Hasil neural network kondisi misalignment	50
Gambar 4.20 Spektrum vibrasi kondisi misalignment	50
Gambar 4.21 Hasil neural network kondisi looseness	51
Gambar 4.22 Spektrum vibrasi kondisi looseness	51



Halaman ini sengaja dikosongkan

RIWAYAT HIDUP



Rahmadi Ridho Anjiko dilahirkan di Nganjuk, 12 Juni 1992. Merupakan putra pertama dari tiga bersaudara pasangan Bapak Sunaryo dan Ibu Siti Samsiyah. Penulis menempuh jenjang pendidikan dari SD Payaman 2 Nganjuk, dan melanjutkan ke SMPN 1 Nganjuk. Kemudian penulis melanjutkan ke SMAN 2 Nganjuk hingga lulus tahun 2011. Pada tahun yang sama penulis melanjutkan studinya ke Institut Teknologi Sepuluh Nopember Surabaya Fakultas Teknologi Industri Jurusan Teknik Elektro dan mengambil bidang studi

Elektronika. Penulis aktif dalam kepanitian acara Electrical Engineering Event yang meliputi LCEN, SITIA, dan Baronas 2014. Penulis juga aktif dalam kegiatan seminar dan pelatihan yang diselenggarakan di lingkungan ITS. Penulis dapat dihubungi melalui email : **ridhoanjiko@gmail.com**



Halaman ini sengaja dikosongkan

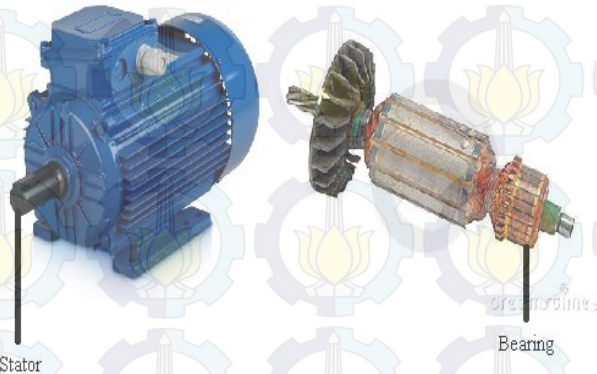
BAB 1

PENDAHULUAN

1.1. Latar Belakang

Dalam kehidupan sehari-hari kita motor induksi banyak digunakan di bidang industri maupun di rumah tangga. Hal ini disebabkan karena motor induksi memiliki konstruksi yang sederhana. Selain karena konstruksinya yang sederhana motor induksi juga mempunyai harga yang relative murah dan mudah dalam pengoperasian dan perawatannya. Motor induksi ini juga mudah didapatkan di toko-toko hardware.

Motor induksi yang umum dipakai adalah motor induksi tiga fasa dan motor induksi satu fasa. Motor induksi tiga fasa dioperasikan pada sistem tiga fasa dan banyak digunakan di industri, sedangkan motor induksi satu fasa dioperasikan pada sistem satu fasa yang banyak digunakan terutama pada penggunaan untuk peralatan rumah tangga seperti pompa air, mesin cuci dan sebagainya. Bentuk fisik dari motor induksi dapat dilihat pada gambar berikut ini



Gambar 1.1 Motor Induksi

Berbagai cara dilakukan untuk memelihara atau merawat motor agar motor selalu dalam keadaan prima. Selain itu juga untuk memperpanjang umur pemakaian motor induksi. Tindakan preventif harus dilakukan untuk mencegah kerusakan yang lebih besar pada

mesin. Salah satu tindakan preventif yang dapat dilakukan adalah monitoring vibrasi pada motor.

Dalam penerapannya di industri base motor akan mengalami pergeseran, namun hal itu tidak akan terjadi secara mendadak, hal ini memerlukan waktu. Sistem vibrasi yang terpasang tidak dapat mengetahui lebih awal jenis atau model kondisi abnormal yang akan terjadi pada motor. Dengan mengetahui lebih awal jenis kondisi abnormal pada pompa maka akan membantu dalam menentukan jenis tindakan pencegahan untuk melindungi pompa dan motor dari kerusakan yang lebih parah.

Dari sisi produksi bila suatu motor tidak dideteksi kondisinya secara kontinyu maka bila suatu saat motor tersebut rusak maka proses produksi perusahaan tersebut akan terganggu. Sehingga otomatis perusahaan tersebut mengalami kerugian.

Hal hal inilah yang melatarbelakangi penulis untuk melakukan penelitian untuk melakukan penelitian ini. Diharapkan dari hasil penelitian ini penulis dapat menemukan suatu sistem tertentu untuk mendeteksi vibrasi atau getaran pada motor induksi sehingga dapat melindungi motor dari kerusakan yang lebih parah. Jika sistem ini ditemukan, maka banyak hal yang dapat dilindungi pada motor diantaranya meningkatkan kehandalan motor, meningkatkan lifetime motor, dan menurunkan biaya pemeliharaan. Algoritma yang akan digunakan adalah fast fourier transform (FFT) dan neural network.

1.2. Perumusan Masalah

Berdasarkan latar belakang di atas, dapat dirumuskan beberapa masalah, antara lain :

1. Bagaimana sensor accelerometer ini mendeteksi vibrasi yang terjadi pada motor induksi.
2. Bagaiman mengetahui pola spektrum vibrasi pada kondisi fault (abnormal) dan unfault (normal).
3. Bagaiman mengimplementasikan sistem ini dalam raspberry.

1.3. Batasan Permasalahan

Untuk menyelesaikan masalah dalam Tugas Akhir ini, maka perlu diberi batasan - batasan dan asumsi sebagai berikut:

1. Membuat program FFT(Fast Fourier Transform) dan Neural Network untuk mendeteksi secara langsung dengan Lazarus di Raspberry Pi, sementara untuk training bobotnya di Lazarus laptop.
2. Penelitian ini hanya mendeteksi motor apakah dalam kondisi unfault, unbalance, misalignment, looseness.

1.4. Tujuan Permasalahan

Tujuan yang ingin dicapai dalam tugas akhir ini adalah sebagai berikut:

3. Untuk mengidentifikasi kerusakan motor melalui spektrum vibrasi yang ditimbulkan.
4. Untuk memonitoring keadaan motor listrik secara kontinyu sehingga keadaan abnormal pada motor listrik dapat diketahui secara dini.

1.5. Metode Penelitian

Metedologi yang digunakan dalam tugas akhir ini adalah sebagai berikut:

5. Studi Literatur
Literatur yang digunakan berasal dari jurnal, prosiding, text book, dan tugas akhir sebelumnya. Dalam studi literatur dipelajari teori pendukung yang berisi tentang pengeretian vibrasi dan bagaimana karakteristik vibrasi, motor listrik dan cara kerjanya, jenis jenis sensor rotating machinery serta cara kerjanya.
6. Perancangan
Penulis menyiapkan motor induksi yang akan dibuat objek penelitian, mencari rumus FFT dan NN yang akan dipakai dalam program yang akan dibuat.
7. Pembuatan Program
Pembuatan program untuk melakukan pendeteksian jenis jenis vibrasi dari motor induksi berdasarkan literatur yang telah dipelajari dan diterapkan pada Lazarus di Raspberry Pi.

8. Analisis Data

Simulasi menggunakan fungsi objektif yang telah ditentukan dan melakukan analisa terhadap jenis jenis kondisi dari motor induksi tersebut.

9. Penulisan Buku

Hasil penelitian yang telah dilakukan dilaporkan dalam bentuk laporan Tugas Akhir dan dikembangkan dalam bentuk paper. Laporan ditulis berdasarkan kesimpulan dari permasalahan yang dianalisis beserta tahapan yang ada di dalamnya.

1.6. Sistematika Penulisan

Untuk mempermudah pembahasan, dalam penulisan tugas akhir ini akan dibagi menjadi lima bab dan beberapa subbab dengan sistematika sebagai berikut:

Bab 1: Pendahuluan

Bab ini membahas mengenai latar belakang, tujuan, rumusan masalah, batasan masalah, manfaat penelitian.

Bab 2: Tinjauan Pustaka

Bab ini membahas teori penunjang mengenai motor induksi, analisa vibrasi, motor induksi, getaran, Raspberry Pi dan Arduino Uno, analisa data pengukuran vibrasi, Fast Fourier Transform (FFT), Neural Network (NN).

Bab 3: Metodologi Penelitian

Bab ini menguraikan sistem identifikasi vibrasi saat melakukan pengukuran on-line, sistem pembelajaran neural network, metode pengambilan data.

Bab 4: Hasil dan Analisa Data

Bab ini berisi tentang pola pola spektrum vibrasi pada motor, klasifikasi neural network.

Bab 5: Kesimpulan dan Saran

Bab ini merupakan bab terakhir yang berisi kesimpulan yang diperoleh dari hasil simulasi dan analisis pada tugas akhir dan saran untuk pengembangan lebih lanjut.

BAB 2

TINJAUAN PUSTAKA

2.1. Analisa Vibrasi

2.1.1. Analisa Vibrasi dan noise pada motor induksi – sumber vibrasi dan metode diagnosanya

Paper ini dibuat Ravindra Birajdar pada konferensi Internasional ke-3 tentang Integrity, Realiability, dan Failure di Portugal pada 20-24 juli 2009. Pada paper ini dijelaskan tentang sumber sumber tentang sumber sumber vibrasi dan noise pada motor dan pompa sentrifugal. Juga diberikan metode diganosa gelombang vibrasi dengan menggunakan Fast Fourier Transform untuk mendapatkan spektrum vibrasi.

Pada penelitian ini, motor yang akan dijadikan objek penelitian adlah motor induksi satu phasa. Metode diagnose yang digunakan adalah sama yaitu dengan menggunakan FFT. Yang pada akhirnya dengan selesainya penelitian ini, penulis dapat membandingkan spektrum vibrasi motor induksi.

2.1.2. Dasar-dasar monitor vibrasi untuk mendeteksi kegagalan dan control proses

Paper ini dibuat oleh Wilfried Reimche di Universitas Hannover, Institut Teknik material di Rio de Janeiro, Brasil pada 2-6 juni 2003. Paper ini menjelaskan bagaimana menganalisa dan memonitor vibrasi pada peralatan-peralatan mekanik yang kompleks misalnya deteksi vibrasi pada turbin uap dan pada motor pada pompa air dengan kapasitas besar (di atas 1000 HP). Analisa yang digunakan juga FFT untuk mendapatkan spektrum vibrasi. Di paper ini tidak disebutkan secara detail jenis pompa atau turbin yang dianalisa.

Sementara pada penelitian ini, analisa vibrasi terfokus motor induksinya.

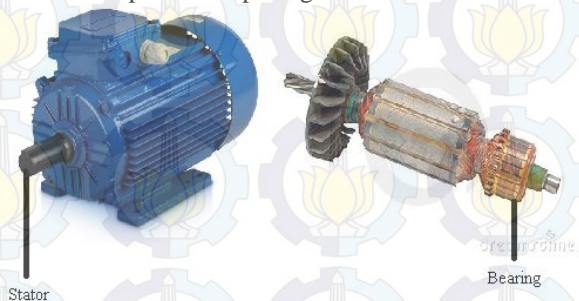
2.1.3. Analizer FFT untuk diagnose kegagalan pada mesin berputar berdasarkan analisa vibrasi

Paper ini dibuat oleh Giovanni Beta (IEEE, 2002) yang melakukan penelitian pada motor listrik tiga phase. Pada penelitian ini, dilakukan pemodelan untuk beberapa jenis kondisi *fault* dan *unfault*.

Pada penelitian ini, pengukuran dilakukan pada motor yang dihubungkan pompa sentrifugal.

2.2. Motor Induksi

Motor induksi sangat banyak digunakan dalam kehidupan sehari-hari baik di industri maupun di rumah tangga karena konstruksinya sederhana, harganya relative murah dan mudah dalam pengoperasian dan perawatan. Motor induksi yang umum dipakai adalah motor induksi tiga phasa dan motor induksi satu phasa. Motor induksi tiga phasa dioperasikan pada sistem tiga phasa dan banyak digunakan di industri, sedangkan motor induksi satu phasa dioperasikan pada sistem satu phasa yang banyak digunakan terutama pada penggunaan untuk peralatan rumah tangga seperti pompa air, mesin cuci dan sebagainya. Bentuk fisik dari motor induksi dapat dilihat pada gambar berikut ini:



Gambar 2.1 Motor Induksi

Motor ini bekerja berdasarkan induksi medan magnet stator ke rotornya. Ketika Tegangan AC 3 phasa yang diberikan ke stator menghasilkan medan putar (*rotating magnetic field*) pada stator. Medan putar tersebut memotong konduktor pada rotor sehingga menghasilkan

induksi ggl (gaya gerak listrik) pada rotor (berdasarkan hukum Faraday). Karena kumparan rotor merupakan rangkaian tertutup maka ggl akan menghasilkan arus I. Adanya arus di dalam medan magnet menimbulkan gaya F pada rotor sehingga rotor berputar.

$$ns = \frac{120f}{p}$$

Dimana :

ns = kecepatan sinkron [rpm]
 f = frekwensi sumber [Hz]
 p = jumlah kutub

Jika motor induksi bekerja pada frekuensi 50 Hz dengan jumlah kutub 2 maka kecepatan putar motor tersebut adalah $ns = 120.50/2 = 3000$ rpm. Perbedaan antara kecepatan sinkron dan kecepatan rotor disebut slip.

$$s = \frac{ns - n}{ns}$$

Dimana :

s = slip
 ns = kecepatan sinkron [r/min]
 n = kecepatan rotor [r/min]

2.3. Getaran

Getaran adalah merupakan gerakan osilasi pada sistem mekanik pada titik referensi. Getaran sendiri dapat dibagi menjadi dua bagian besar yaitu getaran yang periodic maupun acak.

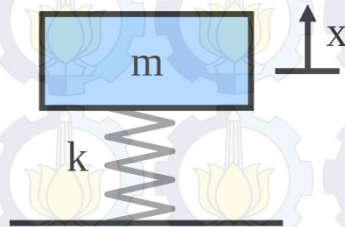
Pada getaran yang berperiodik, maka gerakan akan berulang-ulang pada interval waktu tertentu, demikian yang non periodic tidak berulang-ulang pada interval waktu tertentu. Getaran sendiri dapat dibagi menjadi dua bagian besar yaitu getaran yang periodic maupun acak.

Pada getaran yang berperiodik, maka gerakan akan berulang-ulang pada interval waktu tertentu, demikian yang non periodic tidak berulang-ulang pada interval waktu tertentu.

Getaran pada umumnya ada yang diinginkan ada yang tidak, pada getaran yang diinginkan seperti garputala maupun speaker. Namun

ada getaran yang tidak diinginkan, karena dapat melemahkan daya, akan menimbulkan gangguan suara (Noise), dan mengidentifikasi adanya ketidak normalan dari sistem mekanik yang bergetar.

Prinsip dasar getaran dapat digambarkan sebagai sebuah gerak bebas dari per (Spring). Seperti gambar dibawah ini.



Gambar 2.2 Gerak bebas Spring yang berosilasi

Dimana persamaan untuk gerak bebas sebuah per (spring) adalah: $F_s = -k \cdot x$ dimana persamaan x dapat dituliskan sebagai

$$X(t) = A \sin (2\pi f_n t) = A \sin (\omega t) \quad (2.1)$$

$$f_n = \frac{1}{2\pi} \sqrt{\frac{k}{m}} \quad (2.2)$$

Persamaan 2.1 ini merupakan persamaan pergerakan dari getaran secara umum, maka turunan pertama dari pergerakan getaran adalah merupakan kecepatan dari getaran. Dan turunan kedua dari pergerakan adalah percepatan getaran.

$$v = \frac{dx}{dt} = \omega A \cos(\omega t) \quad (2.3)$$

$$a = \frac{d^2x}{dt^2} = -\omega^2 A \sin(\omega t) \quad (2.4)$$

2.3.1. Penyebab Vibrasi

Identifikasi sumber dan deteksi kerusakan pada sinyal getaran diasosiasikan dengan jenis komponen yang berhubungan dengan getaran rotasi seperti gear, rotor, dan shaft, rolling element bearing, flexible coupling dan mesin listrik.

Penyebab utama dari getaran mekanik adalah ketidakseimbangan (unbalance), ketidaklurusan (misalignment), longgar/kendur (looseness) dan sebagainya.

Vibrasi mekanik sering diukur dengan menggunakan akselerometer, tetapi juga sering menggunakan probe displacement dan sensor velocity. Disamping itu, dapat juga menggunakan analyzer vibrasi.

Hal yang penting adalah untuk mengetahui lokasi sumber vibrasi. Seperti yang dijelaskan di atas bahwa gaya dapat menyebabkan vibrasi. Jika kita mengetahui tipe-tipe gaya yang menyebabkan vibrasi maka kita dapat mengetahui bagaimana gaya-gaya ini akan menyebabkan vibrasi. Pada mesin-mesin berputar, titik-titik vibrasi selalu secara langsung terletak pada bearing mesin. Penyebabnya adalah bahwa berbagai gaya dinamik dari mesin yang berputar harus diambil dalam arah horizontal, vertical dan axial pada masing masing bearing.

Vibrasi pada motor mempunyai pengaruh besar dalam menentukan unjuk kerja dari motor itu sendiri. Pada umumnya, dengan penambahan level vibrasi maka mengidentifikasi adanya kerusakan awal pada motor. Kerusakan awal ini nantinya akan menyebabkan kerusakan fatal pada motor itu sendiri. Vibrasi juga dapat menjadi keluaran dari peralatan yang respon gaya eksitasi, seperti residual rotor unbalance, aliran fluida turbulen, tekanan, kavitasi dan atau wear pompa.

Sumber-sumber vibrasi pada motor dapat dikategorikan dalam tiga tipe yaitu: penyebab mekanik, penyebab hidrolis dan penyebab peripheral.

Penyebab mekanik meliputi:

- Komponen putaran yang tidak seimbang (Unbalanced rotating equipment)
- Impeller yang rusak dan sleeve shaft yang tidak kpsentrik
- Shaft yang bengkok
- Pompa dan motor tidak selurus

- Komponen-komponen pompa dan motor yang berekspansi karena pengaruh kenaikan suhu misalnya shaft
- Bearing yang rusak
- Pipe strain
- Desain pondasi yang tidak tepat
- Penyebab hidrolis meliputi:
 - Pompa beroperasi diluar area titik efisiensi yang terbaik
 - Resirkulasi internal
 - Adanya penguapan
 - Water hammer
 - Adanya udara terjebak dalam sistem
 - Turbelansi dalam sistem
- Penyebab peripheral meliputi:
 - Vibrasi harmonis dari equipment lain yang terdekat
 - Pompa beroperasi pada kecepatan kritis

2.3.2. Accelerometer MEMS (Micro-Electro Mechanical System)

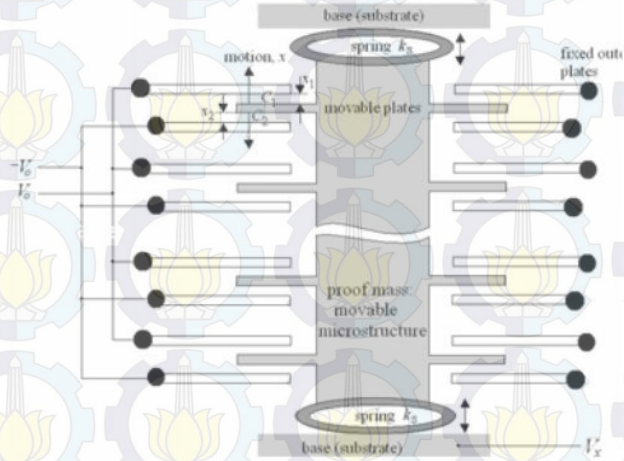
MEMS (Micro-electro Mechanical System) adalah merupakan sebuah sensor mekanik yang dikemas ke dalam bentuk Integrated Circuit (IC). Karena merupakan sebuah microelectronic maka komponen utama penyusunannya adalah silikon dan dalam ukuran micron. Dan bisa digunakan untuk mengukur percepatan, posisi, atau kejutan. Karena material yang digunakan adalah jenis Silikon, maka MEMS dapat dibuat dalam ukuran hingga micron.

Pada sensor percepatan (Accelerometer) konvensional yang ada menggunakan prinsip piezoelectric yang bentuk fisiknya lebih besar, aplikasinya meletakkan sensor ke obyek yang diukur getarannya sedikit lebih susah dan memerlukan daya yang cukup besar. Dengan beberapa kelebihan tersebut diharapkan MEMS dapat menggantikan sensor getaran yang lebih baik.

MEMS untuk percepatan memiliki struktur Capacitance untuk mendeteksi adanya perubahan percepatan (Acceleration). Dimana Capacitance akan berubah-ubah sesuai dengan pergerakan dari getaran sebagai perubahan percepatan.

Dimana ϵ adalah permitifitas bahan, dimana MEMS permitifitas yang digunakan adalah udara sehingga suhu cukup mempengaruhi besarnya permitifitas bahan. A adalah luas permukaan pelat dan d adalah

jarak antar pelat. Gambar spektrum MEMS dapat dilihat pada gambar dibawah ini.



Gambar 2.3 Struktur MEMS (Stephaen,et al, 2004)

Jika percepatan sama dengan nol (tidak ada percepatan), maka $X_1 = X_2 = X_3$ dan nilai $C_1 = C_2$. Dan dimana nilai X_1 dan X_2 dihasilkan dari pergerakan dari pelat pembentuk kapasitansi pada MEMS yang akan berubah-ubah jika adanya percepatan yang timbul pada MEMS. Jika ada percepatan yang muncul maka nilai dari C_1 dan C_2 akan menjadi:

$$C_1 = \epsilon \frac{1}{x_1 + X} \text{ dan } C_2 = \epsilon \frac{1}{x_1 - x} \quad (2.5)$$

Sehingga perbedaan kapasitansinya menjadi:

$$\Delta C = C_1 - C_2 = 2 \epsilon \frac{x}{x^2 - x_1^2} \quad (2.6)$$

Sehingga akan didapat persamaan aljabar menjadi:

$$\Delta C x^2 - 2 \epsilon x - \Delta C x_1^2 = 0 \quad (2.7)$$

Untuk pergerakan percepatan yang terlalu kecil, maka Cx^2 dapat diabaikan. Sehingga persamaan diatas menjadi:

$$x \approx -\frac{x_1^2}{2 \epsilon} \Delta C \quad (2.8)$$

Sehingga dapat dikatakan bahwa nilai pergerakan (x) akan mendekati nilai perbedaan kapasitansi C. dan menurut hukum pegas bahwa gaya F_s proporsional dengan pergerakan x dan sesuai persamaan pegas menjadi:

$$F_s = K.X \quad (2.9)$$

Dimana X merupakan konstanta pegas. Berdasarkan pada hukum Newton kedua maka dapat dituliskan:

$$m.a = m \frac{dx_1^2}{dt^2} = k.x \quad (2.10)$$

sehingga:

$$a = -\frac{k}{m} X \quad (2.11)$$

dari persamaan ini maka akan didapat:

$$a = -\frac{k.x_1^2}{2m\epsilon} \Delta C \quad (2.12)$$

dari sini dapat dikatakan bahwa percepatan (Acceleration) dapat dipengaruhi oleh besarnya perbedaan kapasitansi dari dalam MEMS.

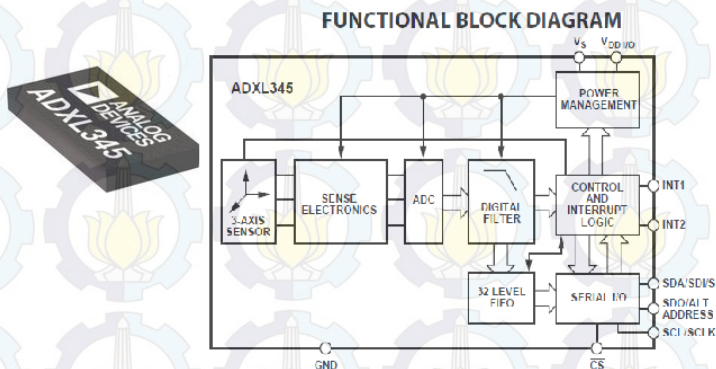
2.3.3. Accelerometer ADXL345 MEMS

Di dunia industry, sensor vibrasi pada umumnya yang digunakan ada dua tipe yaitu tipe proximity dan tipe akselerometer. Tipe proximity digunakan untuk mengukur pergeseran benda-benda yang berputar misalnya pergeseran shaft. Tipe proximity ini tidak bersentuhan langsung dengan obyek yang diukur. Sementara tipe akselerometer ini digunakan untuk mengukur vibrasi benda-benda yang statistic seperti mengukur vibrasi pada rumah bearing. Tipe akselerometer ini bersentuhan langsung dengan obyek yang diukur.

Sensor yang digunakan dalam penelitian ini adalah sensor akselerometer ADXL345. ADXL345 ini memiliki jangkauan pengukuran minimal $\pm 16g$ (gravitasi). Sinyal keluaran tegangan analog yang

dihasilkan oleh sensor ini berbanding lurus dengan percepatan. Akselerometer dapat mengukur percepatan statis gravitasi dalam aplikasi deteksi kemiringan dan juga mengukur percepatan dinamis yang dihasilkan dari gerakan, kejutan/gonjangan atau getaran.

Akselerometer ADXL345 adalah sejenis Micro-Electro-Mechanical Sensor (MEMS). Dari namanya ada hubungan dengan benda mekanik dan sensor elektronik didalamnya. Disebut mikro karena ukuran komponen dalam chips seukuran 0,001 – 0,1 mm. Prinsip kerja yang digunakan adalah prinsip percepatan (acceleration). Sebuah per dengan beban dan diepaskan, beban bergerak dengan suatu percepatan sampai kondisi tertentu akan berhenti. Bila ada sesuatu yang menggoncangkannya maka beban akan berayun kembali. Konsep ini diterapkan pada sebuah chip kecil, dimana per digantikan dengan plat mini yang ujungnya diberi penyangga dan ditengahnya diberi beban. Lain pelat dipasang dalam posisi tetap berhadapan. Saat ada gaya dari luar maka pelat beban akan bergerak mendekat pelat tetap, bila pelat sebagai kapasitor maka besaran kapasitansi chip MEMS. Agar sensor ini bisa mendeteksi 3 dimensi, maka dibutuhkan 3 pasang plat yang dipasang tegak lurus antar masing masing.



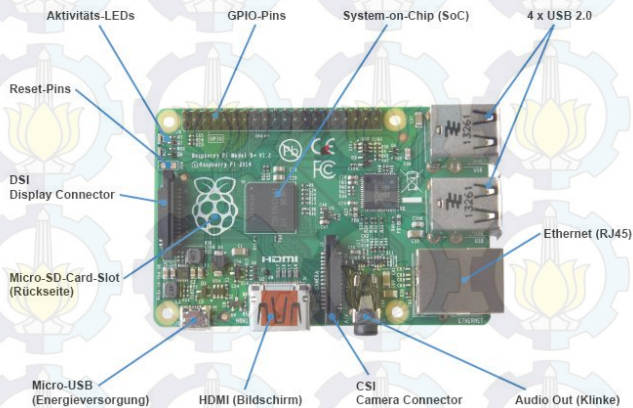
Gambar 2.4 Sensor ADXL345 dan blok diagramnya

2.4. Raspberry Pi

Raspberry Pi (juga dikenal sebagai RasPi) adalah sebuah SBC (Single-Board Computer) seukuran kartu kredit yang dikembangkan

oleh Yayasan Raspberry Pi di Inggris (UK) dengan maksud untuk memicu pengajaran ilmu komputer dasar di sekolah-sekolah.

Raspberry Pi menggunakan system on a chip (SoC) dari Broadcom BCM2835, juga sudah termasuk prosesor ARM1176JZF-S 700 MHz, GPU VideoCore IV dan RAM sebesar 256 MB (untuk Rev. B). Tidak menggunakan hard disk, namun menggunakan SD Card untuk proses booting dan penyimpanan data jangka-panjang.



Gambar 2.5 Raspberry Pi model B+

Specifications

Chip	Broadcom BCM2835 SoC
Core architecture	ARM11
CPU	700 MHz Low Power ARM1176JZF5 Applications Processor
GPU	Dual Core VideoCore IV® Multimedia Co-Processor Provides Open GL ES 2.0, hardware-accelerated OpenVG, and 1080p30 H.264 high-profile decode Capable of 1Gpixel/s, 1.5GTexel/s or 24GFLOPs with texture filtering and DMA infrastructure
Memory	512MB SDRAM
Operating System	Boots from Micro SD card, running a version of the Linux operating system
Dimensions	85 x 56 x 17mm
Power	Micro USB socket 5V, 2A

Gambar 2.6 Spesifikasi Raspberry Pi B+

2.5. Arduino Uno

Arduino Uno adalah papan mikrokontroler berbasis ATmega328. Arduino *Uno* memiliki 14 digital pin input/output, dimana 6 pin digunakan sebagai output PWM, 6 pin *input* analog, 16 MHz *resonator* keramik, koneksi USB, *jack* catu daya eksternal, header ICSP, dan tombol *reset*. Ini semua berisi hal-hal yang diperlukan untuk mendukung mikrokontroler, sederhana saja, hanya dengan menghubungkannya ke komputer dengan kabel USB atau sumber tegangan dengan adaptor AC-DC dan atau baterai untuk memulai menggunakan papan Arduino.

Arduino Uno R3 berbeda dari semua papan Uno sebelumnya yang sudah tidak menggunakan *chip driver* FTDI USB-to-serial. Sekarang, Arduino Uno menggunakan fitur Atmega16U2 (Atmega8U2 sampai dengan versi R2) yang diprogram sebagai konverter USB-to-serial.



Gambar 2.7 Arduino Uno Modul

Arduino Uno dapat diaktifkan melalui koneksi USB atau dengan catu daya eksternal. Sumber daya akan dipilih secara otomatis oleh Arduino. Sumber daya eksternal (non-USB) dapat berasal baik dari adaptor AC-DC atau baterai. Adaptor dapat dihubungkan dengan mencolokkan steker 2,1 mm yang bagian tengahnya terminal positif ke ke *jack* sumber tegangan pada papan. Jika tegangan berasal dari baterai dapat langsung dihubungkan melalui *header* pin Gnd dan pin Vin dari konektor POWER. Papan Arduino Uno dapat beroperasi dengan daya eksternal 6 Volt sampai 20 volt. Jika diberi tegangan kurang dari 7 Volt, maka, pin 5 Volt mungkin akan menghasilkan tegangan kurang dari 5 Volt dan ini akan membuat papan menjadi tidak stabil. Jika sumber

tegangan menggunakan lebih dari 12 Volt, *regulator* tegangan akan mengalami panas berlebihan dan bisa merusak papan. Rentang sumber tegangan yang dianjurkan adalah 7 Volt sampai 12 Volt. Arduino disini saya gunakan untuk membaca sensor ADXL345 yang menggunakan SPI communication, kemudian hasil baca sensornya baru saya olah di raspberry. Langkah ini digunakan karena lazarus di raspberry tidak bisa membaca pin input SPI langsung dari sensor ADXL345, sehingga ditempuh cara ini dengan membuat data menjadi biner terlebih dahulu, karena setelah di konversikan baru bisa dibaca oleh lazarus melalui pin GPIO yang tersedia di raspberry.

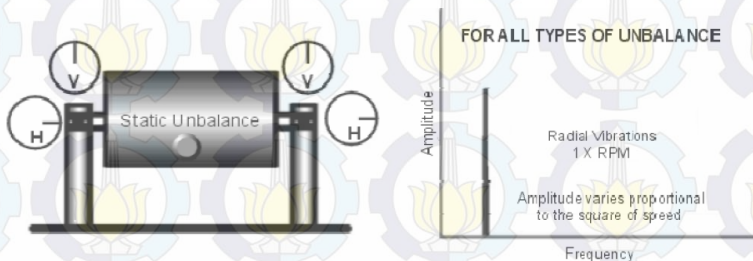
2.6. Analisa Data Pengukuran Vibrasi

Ada berbagai metode yang digunakan dalam menganalisa data-data vibrasi. Metode dasar dengan menampilkan data vibrasi dalam area frekuensi, juga disebut dengan spektrum vibrasi. Frekuensi vibrasi adalah jumlah siklus vibrasi per satuan waktu. Spektrum vibrasi adalah bagian dasar dalam menganalisa vibrasi, karena dapat menampilkan informasi-informasi yang tersembunyi pada bentuk gelombang vibrasi berdasarkan waktu. Untuk mendapatkan spektrum vibrasi dapat dilakukan dengan berbagai cara, yang sangat sering digunakan adalah Fast Fourier Transform (FFT).

Dari grafik dalam bentuk gelombang FFT, dapat didiagnosa penyebab vibrasi dan dapat juga dilihat model model vibrasi. Berikut ini adalah beberapa model vibrasi yang umum terjadi.

2.6.1. Ketidakeimbangan (Unbalance)

Ketidakeimbangan untuk seterusnya akan kita sebut *unbalance*, FFT spektrum akan menunjukan puncak gelombang pada frekuensi 1x rpm vibrasi dan amplitud pada frekuensi 1x rpm akan bervariasi proposional terhadap kuadrat kecepatan putaran. Jika *unbalance* terjadi maka puncak gelombang pada frekuensi 1x rpm mendominasi spektrum vibrasi.

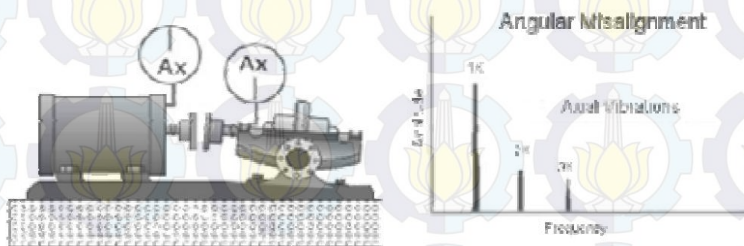


Gambar 2.8 Lokasi pengukuran vibrasi dan pembacaan FFT yang menunjukkan puncak vibrasi karena unbalance (Birajdar, Ravindra, 2009)

2.6.2. Pompa dan Motor Yang Tidak Selurus (Pump and Motor Misalignment)

Istilah tidak selurus selanjutnya akan diganti dengan *misalignment*. Pada dasarnya ada 2 tipe *misalignment* yang terjadi diantara pompa dan motor:

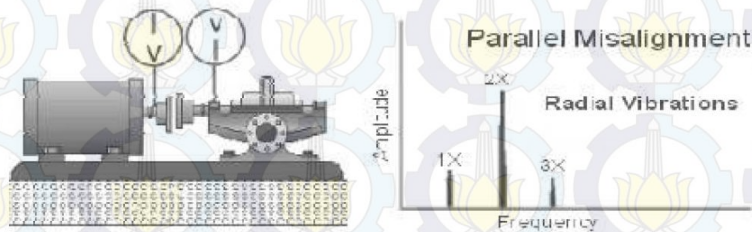
- *Misalignment* angular – yaitu garis tengah kedua shaft (shaft motor dan shaft pompa) membentuk suatu sudut
- *Misalignment* parallel – yaitu garis tengah kedua shaft (shaft motor dan shaft pompa) parallel.



Gambar 2.9 Misalignment angular (Birajdar, Ravindra, 2009)

Seperti yang ditunjukkan pada gambar 2.8 bahwa misalignment angular akan menyebabkan puncak gelombang vibrasi axial pada frekuensi 1x rpm. Secara umum pada misalignment jenis ini, vibrasi axial dapat juga dilihat pada frekuensi 1x rpm, 2x rpm, 3 x rpm.

Untuk misalignment parallel dapat dilihat pada gambar dibawah ini.



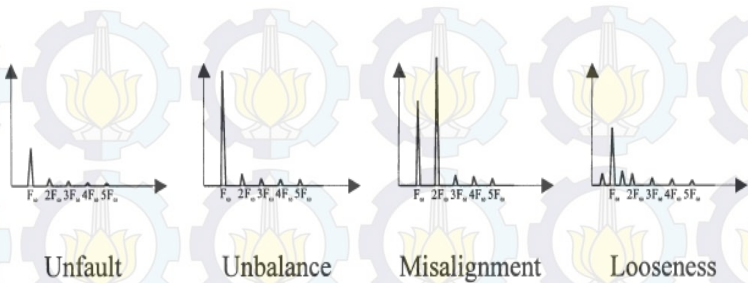
Gambar 2.10 Misalignment Paralel (Birajdar, Ravindra, 2009)

Misalignment parallel akan menghasilkan spektrum vibrasi dengan puncak vibrasi radial pada frekuensi 2x rpm. Vibrasi radial pada frekuensi 1x rpm dan 3x rpm tetap muncul, tetapi lebih kecil dari frekuensi 2x rpm.

2.6.3. Identifikasi Model-model Vibrasi Kondisi *Unfault* dan *Fault*

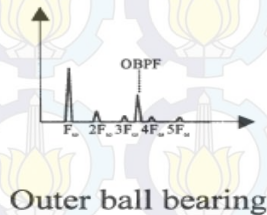
Pendekatan model berbasis diagnostik memerlukan identifikasi dari model *unfault* (normal) dan *faults* (tidak normal). Dalam analisis vibrasi frekuensi-domain, model ini dapat terdiri dari parameter yang sesuai dari spektrum getaran sinyal. Spektrum getaran selama kondisi normal ditandai dengan amplitude tertinggi terletak pada frekuensi putaran motor, diikuti oleh sejumlah frekuensi harmonik yang amplitudanya kurang dari satu-sepertiga amplitude frekuensi putaran (lihat gambar 2.9). Setiap jenis kegagalan mekanis menyebabkan perubahan tertentu dari spektrum vibrasi yang kemudian disebut kondisi tidak normal. Berikut ini, beberapa contoh model vibrasi tidak normal yang paling umum terjadi.

sebuah kondisi *unbalance* (ketidakseimbangan poros) menyebabkan amplitude vibrasi radial tertinggi pada frekuensi putaran. Untuk kondisi *misalignment* amplitude tertinggi akan terjadi pada frekuensi-frekuensi harmonik (dua sampai lima kali frekuensi putaran). Semakin tinggi nilai amplitude pada frekuensi harmonik menunjukkan semakin tinggi tingkat kerusakan kondisi *misalignment*. Sementara ini, untuk kondisi (longgar/kendur), hal ini akan menyebabkan bertambahnya frekuensi harmonik dan akan menyebabkan juga semakin tinggi nilai amplitude pada frekuensi putaran jika dibandingkan dengan kondisi normal.



Gambar 2.11 Spektrum vibrasi untuk kondisi unfault dan beberapa kondisi fault (Giovanni Betta, IEEE, 2002)

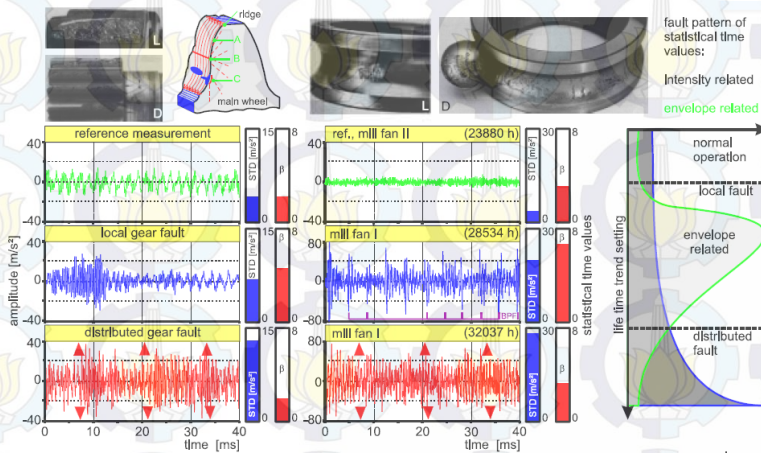
Selain kondisi diatas, kerusakan *bearing* juga akan membangkitkan suatu frekuensi harmonik diluar frekuensi harmonik putaran poros motor. Frekuensi-frekuensi yang akan muncul diantaranya frekuensi kecepatan putaran, *outer* dan *inner ball pass frequencies* (IBPF dan OBPF) dan frekuensi putaran ball. Berikut ini adalah spektrum frekuensi untuk kondisi OBPF.



Gambar 2.12 Spektrum frekuensi kondisi kerusakan bearing pada outer ball pass (OBPF)

2.7. Fast Fourier Transform

Sinyal vibrasi yang diukur oleh transducer masih sebagai fungsi waktu dengan satuan amplitud tertentu. Berikut ini adalah contoh hasil pengukuran fungsi waktu vibrasi.



Gambar 2.13 Sinyal fungsi waktu hasil pengukuran vibrasi

Untuk menganalisa sinyal vibrasi diperlukan dua informasi yaitu amplitudo dan frekuensi. Untuk mendapatkan informasi ini digunakan analisa yang disebut Fast Fourier Transform. FFT ini berfungsi untuk mengubah sinyal dari *time base* menjadi *frequency base*.

Persamaan dasar dari FFT ini adalah :

$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^{kn} \quad (2.13)$$

Dengan:

n, k, N adalah integer

$K = 0, 1, 2, 3, \dots, N-1$

$$j = \sqrt{-1}$$

$$W_N = e^{-j2\pi/N} = \cos \frac{2\pi}{N} - j \sin \frac{2\pi}{N} \quad (2.14)$$

Karena $x(n) = x_r(n) + jx_i(n)$ bisa bernilai kompleks, maka

$x(k) = x_R(k) + jX_I(k)$ yaitu:

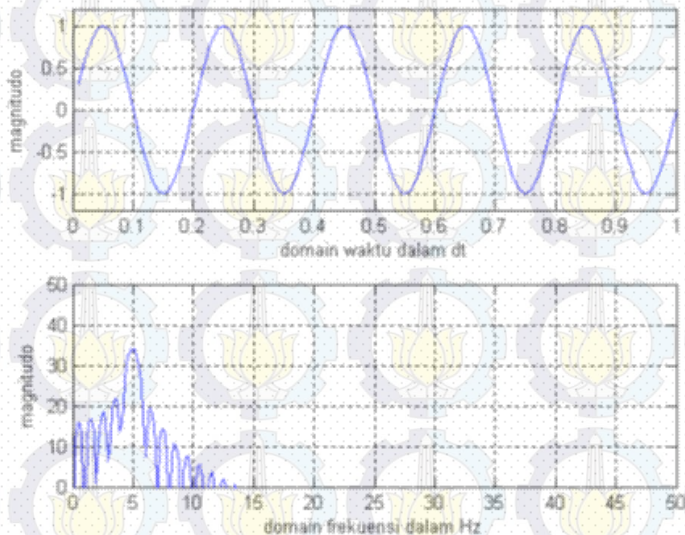
$$X_R(k) = \sum_{n=0}^{N-1} \left[X_r(n) \cos 2\pi \frac{k}{N} n + X_i(n) \sin 2\pi \frac{k}{N} n \right] \quad (2.15)$$

$$X_I(k) = \sum_{n=0}^{N-1} \left[X_r(n) \sin 2\pi \frac{k}{N} n + X_i(n) \cos 2\pi \frac{k}{N} n \right] \quad (2.16)$$

Dari persamaan ini dapat dilihat memerlukan $2N^2$ evaluasi fungsi trigonometric ditambah $4N^2$ perkalian real ditambah $4N(N-1)$ penjumlahan real dan sejumlah *indexing* dan *addressing operators*. Hal ini sering disebut dengan $O(N^2)$.

Persamaan ini merupakan bentuk transformasi fourier yang siap dikomputasi secara langsung dari bentuk sinyal $x(t)$.

Sebagai contoh, jika memiliki sinyal sinus dengan frekuensi 5 Hz dan amplitude 1 volt. Dalam domain waktu akan dapat dilihat seperti pada gambar 1 bagian atas. Sementara dalam domain frekuensi akan didapatkan seperti pada bagaian bawah.



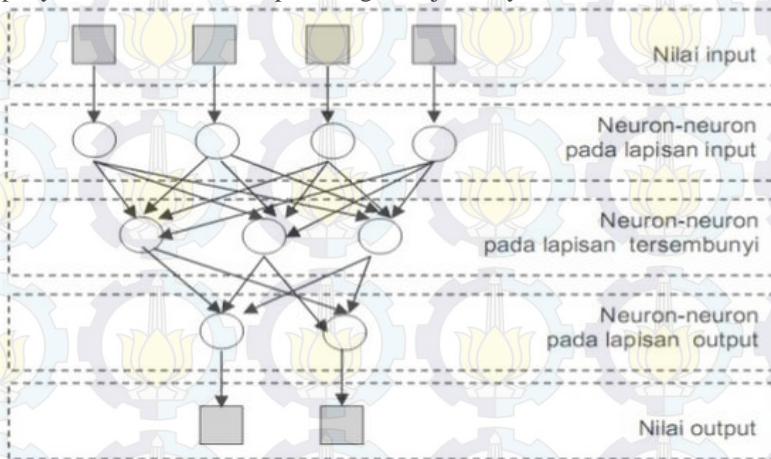
Gambar 2.14 Sinyal sinus dalam domain waktu dan frekuensi

2.8. Neural Network (Jaringan Saraf Tiruan)

Jaringan saraf tiruan (JST) atau neural network adalah suatu metode komputasinya yang meniru sistem jaringan saraf biologis. Metode ini menggunakan elemen perhitungan non-linier dasar yang disebut neuron yang diorganisasikan sebagai jaringan yang saling berhubungan, sehingga mirip dengan jaringan saraf manusia. Jaringan

saraf tiruan dibentuk untuk memecahkan suatu masalah tertentu seperti pengenalan pola atau klasifikasi karena proses pembelajaran.

Setiap pola-pola informasi input dan output yang diberikan kedalam JST diproses dalam neuron. Neuron-neuron tersebut terkumpul di dalam lapisan-lapisan yang disebut neuron layers. Lapisan-lapisan penyusun JST tersebut dapat dibagi menjadi 3, yaitu:



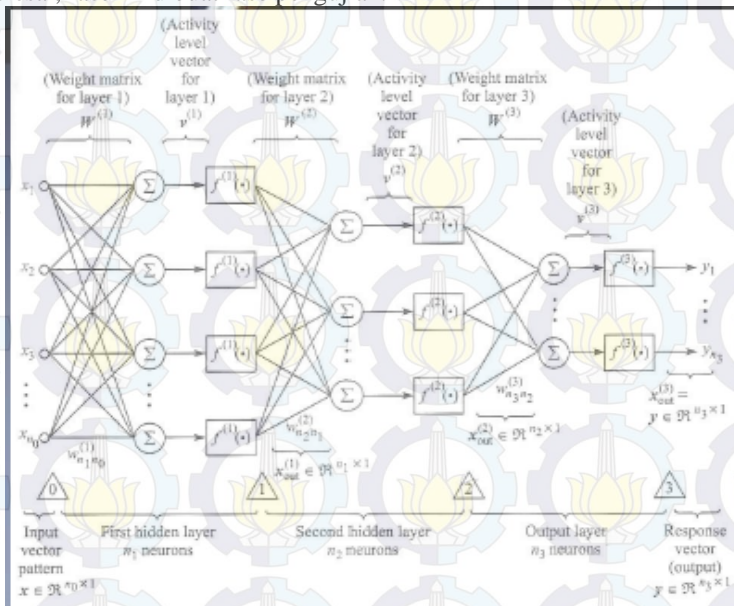
Gambar 2.15 Jaringan saraf dengan 3 lapisan

1. Lapisan Input
Unit-unit di dalam lapisan input disebut unit-unit input. Unit unit input tersebut menerima pola inputan data dari luar yang menggambarkan suatu permasalahan.
2. Lapisan tersembunyi
Unit-unit di dalam lapisan tersembunyi disebut unit unit tersembunyi. Dimana outputnya tidak dapat secara langsung diamati.
3. Lapisan output
Unit-unit di dalam lapisan tersembunyi disebut unit-unit output. Output dari lapisan ini merupakan solusi JST terhadap suatu permasalahan.

Metode algoritma yang digunakan dalam penelitian ini adalah algoritma backpropagation. Propagasi balik atau backpropagation merupakan suatu teknik pembelajaran/pelatihan supervised learning yang paling banyak digunakan. Metode ini merupakan salah satu metode

yang sangat baik dalam menangani masalah pengenalan pola-pola kompleks.

Di dalam jaringan propagasi balik, setiap unit yang berbeda di lapisan input terhubung dengan setiap unit yang ada di lapisan tersembunyi. Setiap unit yang ada di lapisan tersembunyi terhubung dengan setiap unit yang ada di lapisan output. Jaringan ini terdiri dari banyak lapisan (multilayer network). Ketika jaringan diberikan pola masukan sebagai pola pelatihan, maka pola tersebut menuju unit-unit lapisan tersembunyi untuk selanjutnya diteruskan pada unit unit di lapisan keluaran. Kemudian unit unit lapisan keluaran akan memberikan respon sebagai keluaran JST. Saat hasil keluaran tidak sesuai dengan yang diharapkan, maka keluaran akan disebarkan mundur (backward) pada lapisan tersembunyi kemudian dari lapisan tersembunyi menuju lapisan masukan. Tahap pelatihan ini merupakan langkah untuk melatih suatu JST, yaitu dengan cara melakukan perubahan bobot. Sedangkan penyelesaian masalah akan dilakukan jika proses pelatihan tersebut telah selesai, fase ini dibuat fase pengujian.



Gambar 2.16 Arsitektur Backpropagation

Aturan pelatihan jaringan propagasi balik terdiri dari 2 tahapan, feedforward dan backward propagation. Pada jaringan diberikan sekumpulan contoh pelatihan yang disebut set pelatihan. Set pelatihan ini diasosiasikan dengan sebuah vector feature yang disebut dengan vector input yang diasosiasikan dengan sebuah output yang menjadi target pelatihannya. Dengan kata lain set pelatihan terdiri dari vector input dan juga vector output target. Keluaran dari jaringan berupa sebuah vector output actual. Selanjutnya dilakukan perbandingan antara output actual yang dihasilkan dengan output target dengan cara melakukan pengurangan diantara kedua output tersebut. Hasil dari pengurangan merupakan error. Error dijadikan sebagai dasar dalam melakukan perubahan dari setiap bobot yang ada dengan mempropagasikan kembali. Setiap perubahan bobot yang terjadi dapat mengurangi error. Siklus setiap perubahan bobot dilakukan pada setiap set pelatihan hingga kondisi berhenti dicapai, yaitu bila mencapai jumlah bobot yang diinginkan atau hingga sebuah nilai ambang yang ditetapkan terlampaui. Alogaritma pelatihan jaringan propagasi balik terdiri dari 3 tahapan yaitu:

1. Tahap umpan maju (feedforward)
2. Tahap umpan mundur (backpropagation)
3. Tahap pengupdaten bobot.

Alogaritma standart backpropogation yang digunakan adalah sebagai berikut:

$$w_{ji}^{(s)}(k+1) = w_{ji}^{(s)}(k) + \mu^{(s)} \delta_j^{(s)} x_{out,i}^{(s-1)} \quad (2.17)$$

dengan,

$$\delta_j^{(s)} = (d_{qh} - x_{out,i}^{(s)}) g(v_j^{(s)}) ; \text{ untuk layer terluar} \quad (2.18)$$

$$\delta_j^{(s)} = \left(\sum_{h=0}^{s+1} \delta_h^{(s+1)} w_{hj}^{(s+1)} \right) g(v_j^{(s)}) \quad (2.19)$$

BAB 3

PERANCANGAN SISTEM

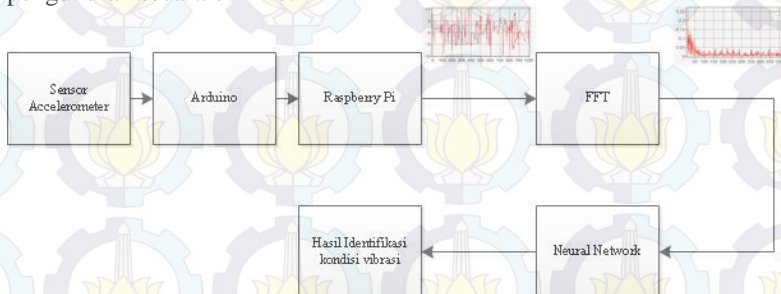
Pada penelitian ini, plant motor yang menjadi obyek penelitian merupakan plant yang berjalan kontinyu tanpa berhenti. Sehingga dibuat suatu prototype motor yang dilengkapi oleh sensor vibrasi. Dengan prototype ini dapat diteliti pola spektrum vibrasi yang terjadi motor induksi.

Metode penelitian ini dibagi dua yaitu:

1. Sistem identifikasi vibrasi saat melakukan pengukuran on-line
2. Sistem pembelajaran neural network

3.1. Sistem Identifikasi Vibrasi saat melakukan pengukuran on-line

Gambar 3.1 adalah blok diagram sistem vibrasi saat melakukan pengukuran secara on-line.



Gambar 3.1 Blok diagram sistem vibrasi pengukuran on-line

Penjelasan untuk masing masing bagian blok diagram diatas seperti dibawah ini.

3.1.1. Sensor Accelerometer

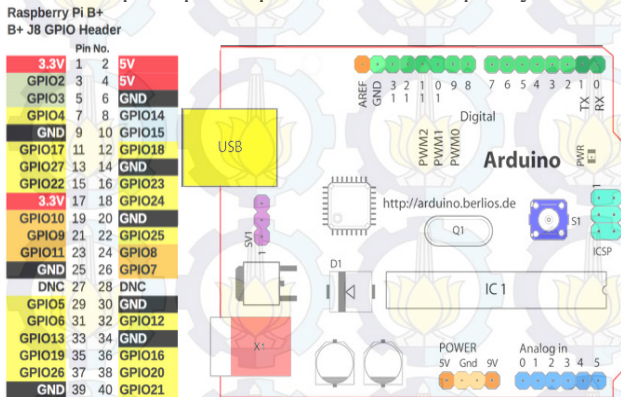
Sensor yang digunakan dalam penelitian ini adalah accelerometer ADXL345. Sensor ini mengukur percepatan dari suatu peralatan. Keluaran dari sensor berupa tegangan DC dengan jangkauan 0-3,3 VDC. Dari data sheet sensor (lampiran A), setiap 256 LSB keluaran sensor setara dengan 1g ($9,8 \text{ m/s}^2$). Sehingga:

$$1 \text{ lsb} = \frac{9,8 \text{ m/s}^2}{256} = 0.0383 \text{ m/s}^2$$

Sehingga dari perhitungan diatas dapat dilihat bahwa setiap 1 LSB setara dengan percepatan 0,0383 m/s². Sensor adxl 345 ini bisa menggunakan SPI (Serial Peripheral Interface) untuk proses komunikasinya. Di sini saya menggunakan SPI untuk komunikasi sensor dengan raspberry pi.

3.1.2. Raspberry Pi

Sistem yang digunakan untuk membaca sensor accelerometer adalah, kita terlebih dahulu menghubungkan sensor tersebut ke arduino, ini dikarenakan, sensor accelerometer tidak mau dibaca secara langsung dengan oleh lazarus yang ada di raspberry pi. Tapi lazarus raspberry pi dapat mendeteksi inputan dari GPIO. Oleh karena itu kita lewatkan data yang akan di baca di lazarus raspberry dengan terlebih dahulu mengubah bentuk data yang dari asalnya float dilewatkan ke arduino diubah ke biner, sehingga raspberry dapat membaca data dari accelerometer tersebut. Gambar pin input output arduino dan raspberry:



Gambar 3.2 GPIO Raspberry dan Arduino

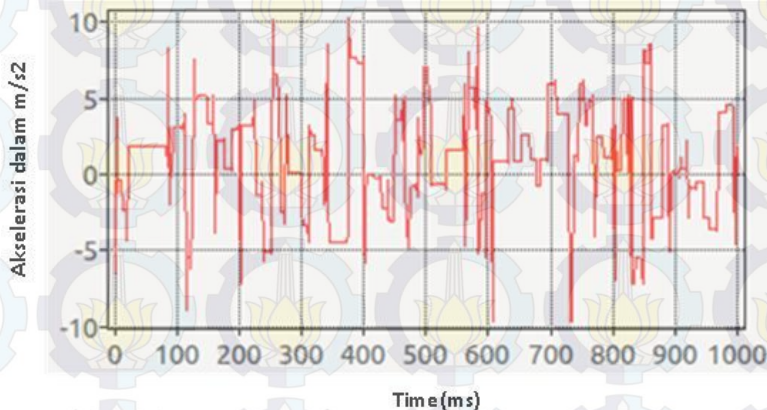
Cara yang digunakan adalah menghubungkan pin 2 sampai 13 di arduino dengan GPIO 0 sampai 12 yang tersedia di raspberry. Maka pin pin tersebut memuat nilai biner yang terbaca dari sensor accelerometer. Sehingga nilai yang terbaca tersebut dapat terbaca di lazarus raspberry. Di lazarus raspberry harus menggunakan library rpi_hal-Hardware Abstraction Library (GPIO, I2C, and SPI) function and procedures.

Library ini diciptakan oleh Stefan Fischer, library ini memuat prosedur dan fungsi untuk mengakses rpi HW I2C, SPI, dan GPIO. Fungsi dan procedure untuk GPIO yang ada di library ini:

```
procedure gpio_set_pin (pin:longword;highlevel:boolean); { Set RPi  
GPIO pin to high or low level; Speed @ 700MHz -> 0.65MHz }  
function gpio_get_PIN (pin:longword):boolean; { Get RPi GPIO pin  
Level is true when Pin level is '1'; false when '0'; Speed @ 700MHz ->  
1.17MHz }  
procedure gpio_set_input (pin:longword); { Set RPi GPIO pin to input  
direction }  
procedure gpio_set_output(pin:longword); { Set RPi GPIO pin to output  
direction }  
procedure gpio_set_alt (pin,altfunc:longword); { Set RPi GPIO pin to  
alternate function nr. 0..5 }  
procedure gpio_set_gppud (mask:longword); { set RPi GPIO Pull-  
up/down Register (GPPUD) with mask }
```

3.1.3. FFT (Fast Fourier Transform)

Nilai keluaran yang dibaca raspberry tersebut akan di baca dengan pemrograman lazarus. Sinyal yang di baca oleh raspberry adalah berupa sinyal dalam fungsi waktu (domain waktu). Hasil keluaran ini ditampilkan dalam bentuk grafik dibawah ini.



Gambar 3.3 Sinyal domain waktu keluaran accelerometer

Selanjutnya dengan FFT, sinyal waktu akan dikonversikan ke domain frekuensi. Persamaan dasar FFT ini adalah:

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn} \quad (3.1)$$

Dimana:

N , k , N adalah integer

$k = 0, 1, 2, 3, \dots, N-1$

$$j = \sqrt{-1}$$

$$W_N = e^{-j2\pi/N} = \cos \frac{2\pi}{N} - j \sin \frac{2\pi}{N} \quad (3.2)$$

Sehingga:

$$W_N^{kn} = e^{-j2\pi kn/N} = \cos \frac{2\pi kn}{N} - j \sin \frac{2\pi kn}{N} \quad (3.3)$$

Karena $x(n) = x_r(n) + jx_I(n)$ bisa bernilai kompleks, dan $X(k) = X_R(k) + jX_I(k)$ maka persamaan (3.1) menjadi :

$$\begin{aligned} X(k) &= \sum_{n=0}^{N-1} x(n) W_N^{kn} = \\ &= \sum_{n=0}^{N-1} \left[(x_r(n) + jx_I(n)) \left(\cos \frac{2\pi kn}{N} - j \sin \frac{2\pi kn}{N} \right) \right] = \\ &= \sum_{n=0}^{N-1} \left[\left(x_r(n) \cos \frac{2\pi kn}{N} - x_r(n) j \sin \frac{2\pi kn}{N} + \right. \right. \\ &\quad \left. \left. jx_I(n) \sin \frac{2\pi kn}{N} - j^2 x_I(n) \cos \frac{2\pi kn}{N} \right) \right] \end{aligned}$$

$$\begin{aligned} X(k) &= \sum_{n=0}^{N-1} \left[\left(x_r(n) \cos \frac{2\pi kn}{N} - x_I(n) \sin \frac{2\pi kn}{N} - \right. \right. \\ &\quad \left. \left. j(x_r(n) \sin \frac{2\pi kn}{N} - x_I(n) \cos \frac{2\pi kn}{N}) \right) \right] \end{aligned}$$

Karena $X(k) = X_R(k) + jX_I(k)$, maka:

$$X_R(k) = \sum_{n=0}^{N-1} \left[x_r(n) \cos \frac{2\pi kn}{N} - x_I(n) \sin \frac{2\pi kn}{N} \right] \text{ (real)} \quad (3.4)$$

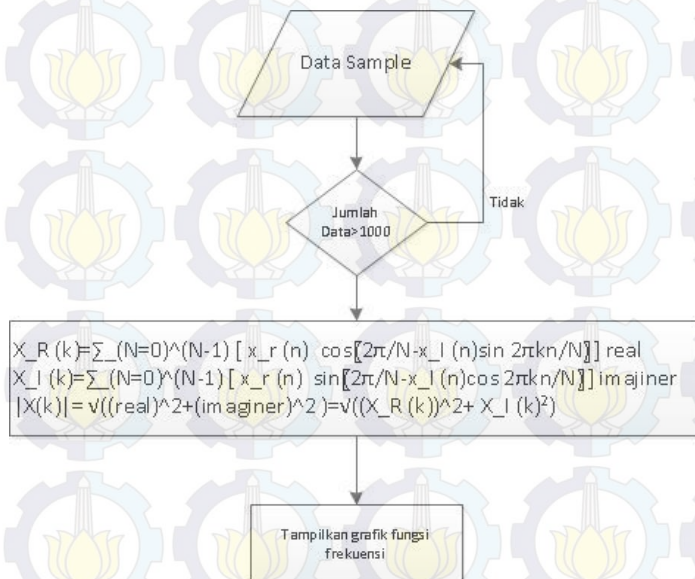
$$X_I(k) = \sum_{n=0}^{N-1} \left[x_r(n) \sin \frac{2\pi kn}{N} - x_I(n) \cos \frac{2\pi kn}{N} \right] \text{ (imajiner)} \quad (3.5)$$

Magnitude pada domain frekuensi dapat dihitung berdasarkan persamaan berikut ini:

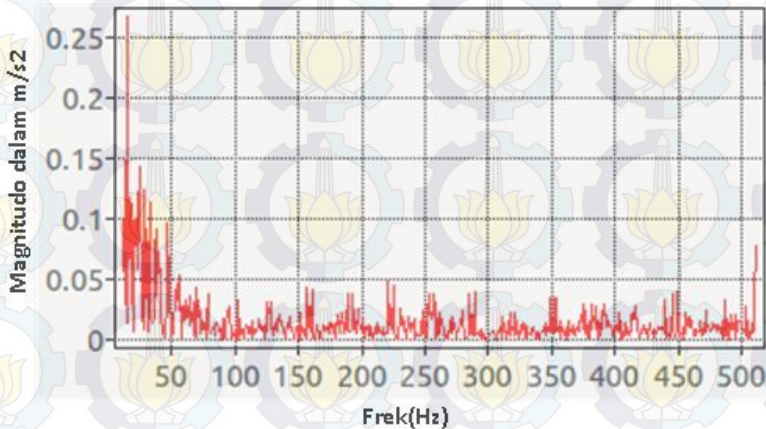
$$|X(k)| = \sqrt{(\text{real})^2 + (\text{imaginer})^2} = \sqrt{(X_R(k))^2 + X_I(k)^2} \quad (3.6)$$

Proses FFT dilakukan setiap 1000 data masukan domain waktu. Adapun flowchart untuk proses FFT adalah seperti gambar 3.4.

Setelah dilakukan proses FFT, kemudian akan ditampilkan grafik dalam fungsi frekuensi seperti gambar 3.5. Dalam domain frekuensi jika semua data frekuensi sebelah kanan sebagai pencerminan frekuensi sebelah kiri, dalam hal ini yang ditampilkan bagian kanan saja.



Gambar 3.4 Flow chart proses FFT dengan sample data vibrasi 1000 data

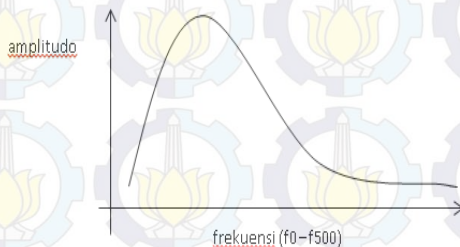


Gambar 3.5 Grafik sinyal dalam domain frekuensi

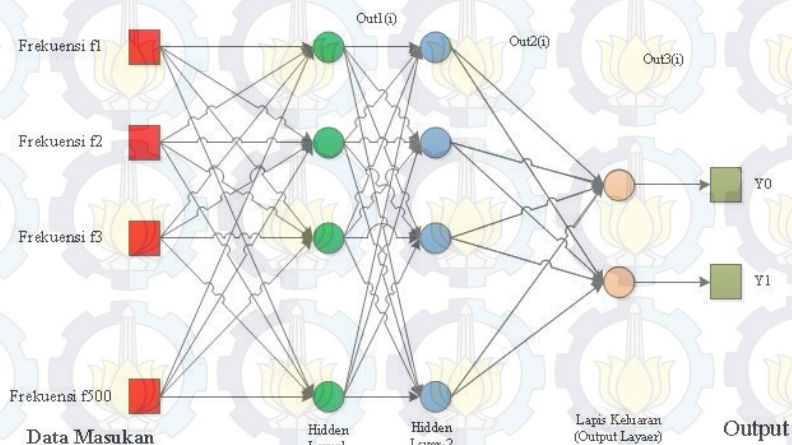
3.1.4. Neural Network

Setelah dilakukan proses FFT maka akan menghasilkan data frekuensi (f_0 sampai f_{500}). Data data frekuensi ini kemudian akan dilewatkan melalui proses neural network untuk menentukan pola vibrasi yang terjadi. Jika digambarkan secara umum grafik frekuensi akan tampak seperti gambar 3.6.

Dari grafik 3.6 tampak bahwa ada nilai frekuensi (f_0 sampai f_{500}). Nilai ini kemudian akan menjadi data masukan pada neural network. Adapun arsitektur neural network pada penelitian ini untuk pembacaan online adalah seperti gambar 3.7.



Gambar 3.6 Grafik domain frekuensi



Gambar 3.7 Arsitektur neural network dalam penelitian ini

Untuk nilai keluaran neural network (target) akan mengacu kepada table 3.1.

Tabel 3.1 Tabel hubungan pola vibrasi dengan NN

Pola Vibrasi	Target	
	Y0	Y1
Looseness (Longgar/Kendur)	0	0
Misalignment (Ketidaklurusan)	0	1
Unbalance (Ketidakseimbangan)	1	0
Unfault (Normal)	1	1

Persamaan yang digunakan dalam neural network ini pada setiap layer adalah sebagai berikut.

Persamaan umum:

$$Output = \frac{1}{1 + e^{-y}} \quad (3.7)$$

$$y = \left(\sum_{i=1}^N w(i) * x(i) \right) \quad (3.8)$$

Untuk hidden layer 1:

$$Output1(k) = \frac{1}{1 + e^{-y1(k)}} \quad (3.9)$$

$$y1(k) = \left(\sum_{i=1}^{500} w(k,i) * f(i) \right) \quad (3.10)$$

Dimana:

k = jumlah neurom hidden layer 1

f(i) = frekuensi masukan ke-i

w1(k,i) = bobot hidden layer 1

Untuk hidden layer 2:

$$Output2(j) = \frac{1}{1 + e^{-y2(j)}} \quad (3.11)$$

$$y2(j) = \left(\sum_{i=1}^k w2(i,j) * output1(i) \right) \quad (3.12)$$

Dimana:

j = jumlah neurom hidden layer 2

w2(I,j) = bobot hidden layer 2

Untuk hidden layer 3:

$$Output3(h) = \frac{1}{1 + e^{-y3(h)}} \quad (3.13)$$

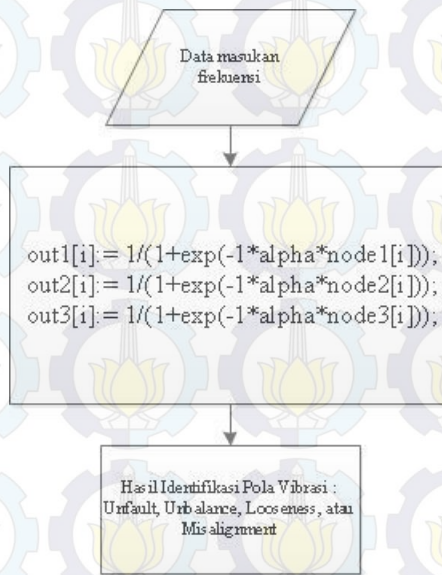
$$y3(h) = \left(\sum_{i=1}^j w3(i,h) * output2(i) \right) \quad (3.14)$$

Dimana:

h = jumlah neurom output layer

w3(i,h) = bobot output layer

Untuk nilai-nilai bobot yang digunakan adalah hasil pembelajaran yang dilakukan dengan lazarus yang ada di komputer. Nilai bobot yang nantinya digunakan merupakan hasil dari nilai training dengan menggunakan backpropagation dimana nilai bobot tersebut setelah mendapatkan hasil yang diinginkan dimana hasil yang diinginkan mempunyai nilai error 0,001. Berikut adalah blok diagram proses neural network pada pengukuran online. Yang dimaksud pengukuran secara online disini adalah pengukuran secara langsung dimana ketika sensor menangkap vibrasi kemudian di teruskan ke lazarus sehingga di prose untuk mengetahui sensor sedang dalam kondisi apa, apakah unfault, unbalance, misalignment ataupun looseness.



Gambar 3.8 Flow chart proses neural network untuk identifikasi pola vibrasi

3.2. Sistem Pembelajaran Nural Network

Proses pembelajaran neural network untuk mendapatkan nilai nilai bobot yang tepat dilakukan secara terpisah menggunakan Lazarus. Langkah langkah yang dilakukan adalah sebagai berikut:

1. Melakukan pengambilan data-data untuk ke-4 kondisi (unfault, unbalance, looseness, misalignment).
2. Pembelajaran dengan lazarus menggunakan data data frekuensi yang sudah didapat.

3.2.1. Pengambilan data frekuensi untuk ke-4 pola vibrasi

Pada tahap ini dilakukan pengambilan data untuk setiap pola vibrasi. Satu paket data frekuensi berisi 500 data (f0-f500). Untuk masing-masing pola vibrasi diambil paket data sebanyak mungkin agar

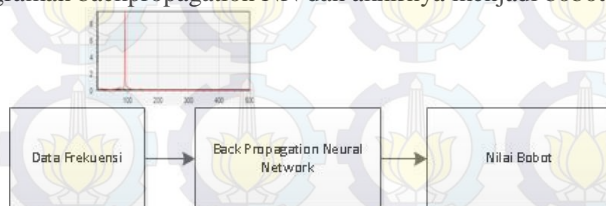
hasil proses pembelajaran dapat mendeteksi pola vibrasi dengan tingkat ketelitian yang lebih tinggi (mendekati 100%).

Cara cara untuk mendapatkan data sesuai dengan pola vibrasi akan dijelaskan pada bagian 3.3. Hasil dari pemngambilan data ini kemudian akan dimasukkan dalam pembelajaran di lazarus.

3.2.2. Pembelajaran dengan Lazarus

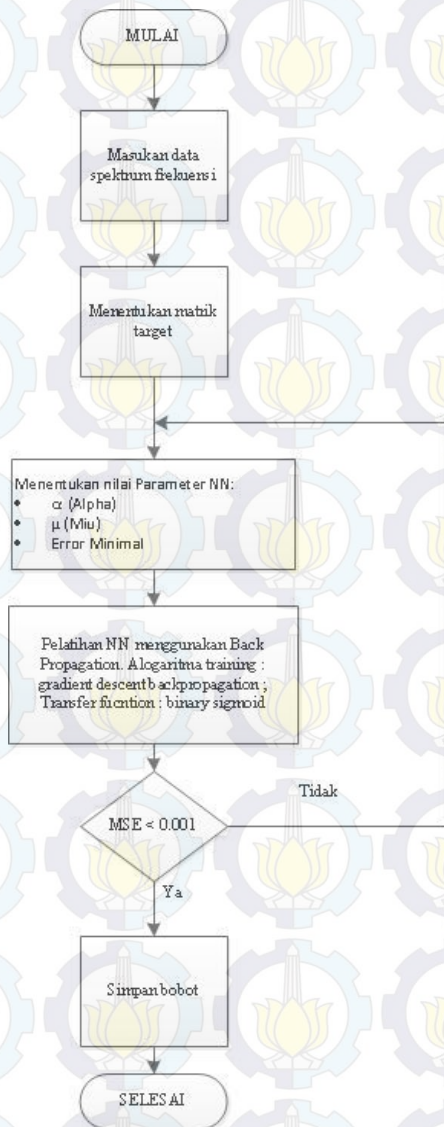
Gambar dibawah alah blok diagram untuk proses pembelajaran neural network.

Dari gambar tersebut dapat dilihat bahwa data-data frekuensi yang akan dijadikan sebagai acuan pembelajaran simasukan dalam pemrograman backpropagation NN dan akhirnya menjadi bobot.



Gambar 3.9 Blok diagram proses pembelajaran NN dengan Lazarus

Adapun flow chart pembelajaran adalah seperti pada gambar 3.10.



Gambar 3.10 Flow chart proses pembelajaran NN dengan Lazarus

Dari proses pembelajaran ini diperoleh nilai nilai bobot pada masing masing neuron. Berikut ini adalah langkah langkah dalam melakukan proses pembelajaran dengan lazarus:

1. Menentukan data yang akan detraining, dalam penelitian ini berupa data-data frekuensi. Data-data training akan meliputi untuk ke-4 pola vibrasi yaitu unfault, unbalance, looseness, dan misalignment. Dalam penelitian ini untuk setiap pola vibrasi akan diambil minimal 5 (lima) data training.
2. Menentukan nilai matrik target untuk masing masing pola vibrasi.
3. Menentukan nilai α (alpha), μ (miu) dan error minimal yang diinginkan.
4. Menentukan jenis transfer fuction untuk masing masing hidden layer dan layer output. Dalam penelitian ini akan menggunakan transfer fuction binary sigmoid $f(Vq) = 1 / (1 + \exp(-n))$
5. Menentukan jenis training. Dalam penelitian ini akan menggunakan suatu fungsi pembelajaran dengan metode backpropagation. Pembelajaran ini dimulai ketika output yang dihasilkan oleh feedforward tidak sesuai dengan yang output harapkan. Perbandingan output ini didapatkan dengan cara menghitung error pada layer output menggunakan persamaan (3.15) dan pada semua layer dengan persamaan (3.16). Setelah itu bobot nilai pada jaringan saraf buatan akan diperbarui dengan konstanta learning rate (μ) berdasarkan persamaan (3.17).

$$\delta_j^{(\ell)} = (d_{qh} - x_{out,j}^{(\ell)})g(v_j^{(\ell)}) \quad (3.15)$$

$$\delta_j^{(\ell)} = \left(\sum_{h=1}^{\ell+1} \delta_h^{(s+1)} w_{hj}^{(\ell+1)} \right) g(v_j^{(\ell)}) \quad (3.16)$$

$$w_{ji}^{(\ell)}(k+1) = w_{ji}^{(\ell)}(k) + \mu \delta_j^{(\ell)} x_{out,j}^{(\ell-1)} \quad (3.17)$$

Setelah mengambil nilai bobot pada neural network, selanjutnya data tersebut disimpan dengan perintah save di lazarus komputer.

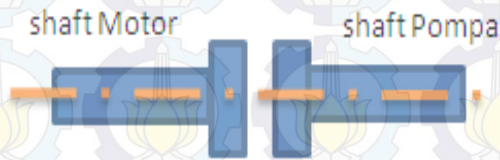
Kemudian nilai-nilai bobot ini digunakan dalam identifikasi pola vibrasi secara online.

3.3. Metode Pengambilan Data

Untuk kondisi kondisi base motor dan pompa pada saat pengambilan data pada penelitian ini saya sesuaikan dengan kondisi base motor dan pompa yang telah dilakukan oleh Moh. Ishak dalam jurnal prosedingnya yang berjudul “Analisis Proteksi Vibrasi Pada Pompa Sentrifugal Dengan Menggunakan Fast Fourier Transform Dan Neural Network” yang diterbitkan pada tahun 2009.

3.3.1. Model kondisi Unfault (Kondisi normal)

Kondisi Unfault atau kondisi normal didefinisikan sebagai kondisi normal operasi motor dan motor dapat beroperasi secara kontinyu. Salah satu ciri kondisi ini yaitu kopling pompa dapat dengan mudah diputar secara manual dengan tangan atau bisa juga dengan melepas beban. Gambar dibawah ini merupakan gambar hubungan kopling motor dan pompa pada saat kondisi unfault.

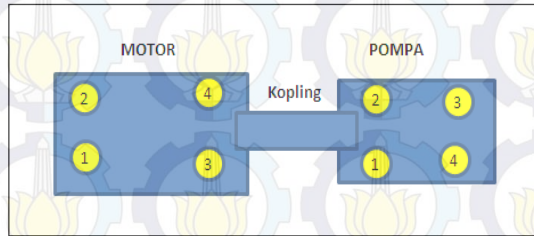


Gambar 3.11 Hubungan Kopling pompa dan motor saat Unfault

3.3.2. Model kondisi Unbalance (Ketidakseimbangan)

Kondisi unbalance yaitu kondisi dimana terjadi ketidakseimbangan pada motor ketika berputar.

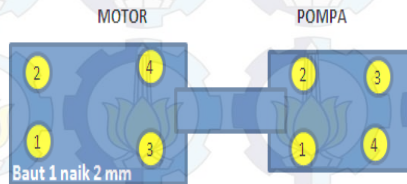
Berikut ini adalah gambar posisi-posisi baut pada base pompa dan motor pada prototipe penelitian ini.



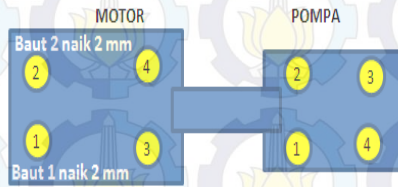
Gambar 3.12 Posisi baut pada prototype base pompa dan motor

Pada gambar diatas, angka angka 1-4 menunjukan posisi baut pada base motor dan pompa dimana pompa dan motor masing masing mempunyai 4 baut. Motor akan dikondisikan *unbalance* dengan cara menaikan satu atau dua titik pada base motor. Hal ini dilakukan dengan cara pengaturan putaran baut.

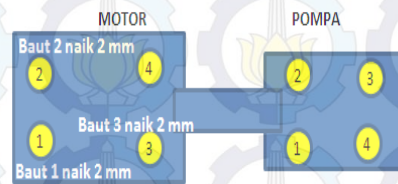
- Kondisi 1 : Base motor dinaikan pada baut no.1 sebanyak putaran baut (2mm)
- Kondisi 2 : Base motor dinaikan pada baut no.1 sebesar a putaran baut (2mm) dan pada baut no.2 sebanyak 1 putaran baut (2mm)
- Kondisi 3 : Base motor dinaikan pada baut no.1 sebesar 1 putaran baut (2mm) dan pada baut no.2 sebesar 1 putaran baut (2mm) dan pada no.3 sebesar 1 putaran baut (2mm)
- Kondisi 4 : Base motor dinaikan pada baut no.1 sebesar 1 putaran baut (2mm), pada baut no.2 sebesar 1 putaran baut (2mm), pada baut no.3 sebesar 1 putaran baut (2mm) dan pada baut no.4 sebesar 0.5 putaran baut (1mm)



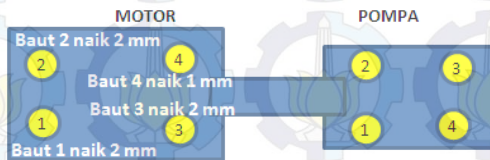
Gambar 3.13 Base motor dan pompa saat Unbalance kondisi 1



Gambar 3.14 Base motor dan pompa saat Unbalance kondisi 2



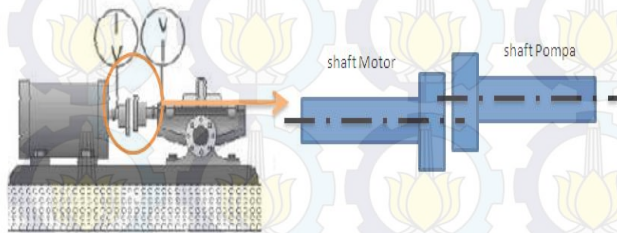
Gambar 3.15 Base motor dan pompa saat Unbalance kondisi 3



Gambar 3.16 Base motor dan pompa saat Unbalance kondisi 4

3.3.3. Model kondisi Misalignment (Tidak Selurus)

Kondisi ini didapat dengan cara menurunkan base motor dari kondisi normal. Kondisi ini ditandai dengan kondisi kopling sangat sulit untuk diputar, ini disebabkan karena kondisi hubungan kopling motor dan pompa tidak selaras. Motor pun untuk memutar kopling pompa melakukan usaha lebih, sehingga jika kondisi ini terus menerus terjadi dapat mengakibatkan kerusakan pada motor dan pompa dan kopling dikarenakan motor harus mengeluarkan tenaga lebih sehingga kecepatan motornya pun lama kelamaan menurun.



Gambar 3.17 Model kondisi Misalignment (Tidak Selurus)

3.3.4. Model kondisi Looseness (Kondisi base kendur)

Dalam penelitian ini, berikut kondisi untuk mendapatkan looseness.

- Kondisi 1: Semua baut base motor dan semua baut base pompa dalam keasaan kendur/longgar.
- Kondisi 2: Semua baut base motor dalam keadaan kendur/longgar, sementara semua baut base pompa dipasang kuat.

BAB 4

HASIL DAN ANALISA DATA

4.1. Pola-pola Spektrum Vibrasi pada Prototipe Motor

Induksi

Prototipe pada penelitian ini menggunakan motor dan pompa yang memiliki kecepatan putaran sebesar 2556 rpm (putaran per menit). Sehingga frekuensi putaran motor adalah sebagai berikut:
frekuensi putaran poros / frekuensi dsar = $\text{rpm}/(60 \text{ detik}) = 2556/60 = 42,6 \text{ Hz}$.

Sementara itu frekuensi-frekuensi harmoniknya adalah sebagai berikut:

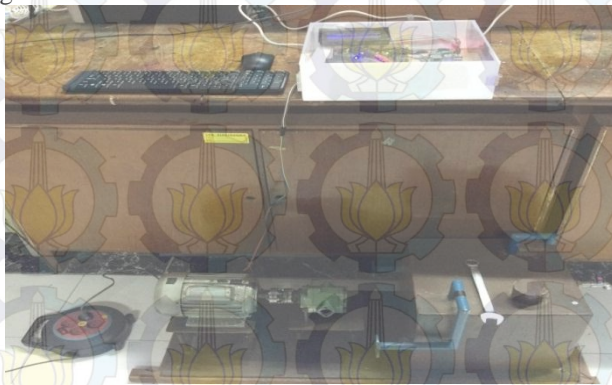
frekuensi harmonik-1 = $42,6 \times 2 = 85,4 \text{ Hz}$

frekuensi harmonik-2 = $42,6 \times 3 = 127,8 \text{ Hz}$

frekuensi harmonik-3 = $42,6 \times 4 = 170,4 \text{ Hz}$

frekuensi harmonik-4 = $42,6 \times 5 = 213 \text{ Hz}$

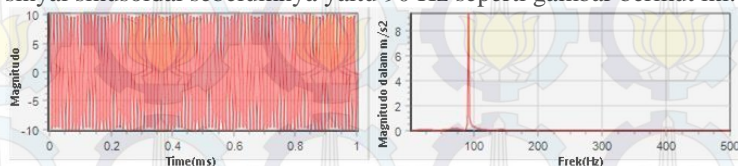
Kemudian setelah dilakukan pembuatan perangkat keras dan perangkat lunak, dilakukan perkaitan dan penyambungan prototipe seperti gambar dibawah ini.



Gambar 4.1 Foto rangkaian prototipe saat pengambilan data

Data yang dihasilkan oleh akselerometer masih berupa domain waktu, yang kemudian melalui program lazarus di raspberry data tersebut diolah menggunakan fast fourier transform menjadi domain

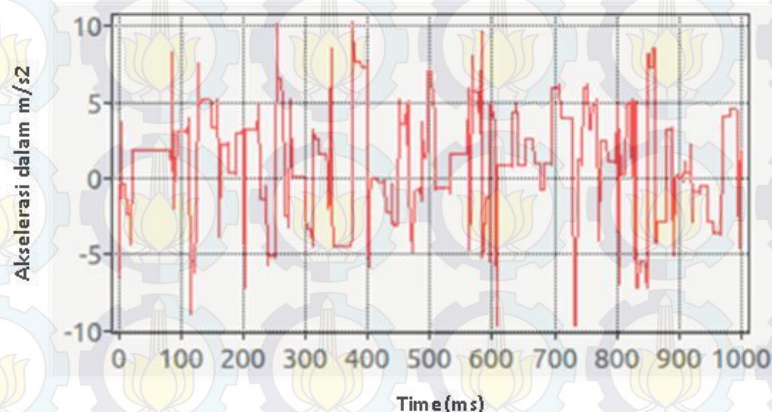
frekuensi. Untuk membuktikan bahwa pemrograman dan perhitungan FFT sudah benar atau tidak maka dilakukan pengujian dengan mensimulasikan sinyal sinusoidal. Sinyal sinusoidal yang saya berikan memiliki frekuensi 90 Hz. Hasilnya menunjukkan bahwa frekuensi yang dominan dalam hasil FFT sama dengan frekuensi yang kita inputkan di sinyal sinusoidal sebelumnya yaitu 90 Hz seperti gambar berikut ini.



Gambar 4.2 Tampilan sinyal pembuktian FFT

Pada penelitian ini, pengukuran vibrasi menggunakan 1 (satu) unit sensor akselerometer Adxl345. Sensor ini mempunyai nilai keluaran voltage, yang kemudian di konversikan menjadi g (gravitasi), dimana $1\text{ g} = 9,8\text{ m/s}^2$. Dari datashet Adxl345, disebutkan bahwa setiap keluaran tegangan 256 lsb setara dengan 1 g.

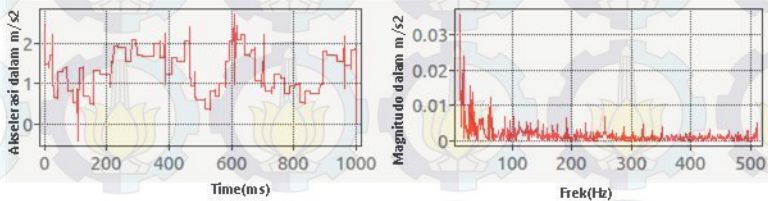
Sinyal yang ditampilkan sebagai hasil pengukuran adalah domain waktu dan domain frekuensi. Berikut ini adalah contoh domain waktu kondisi unfault.



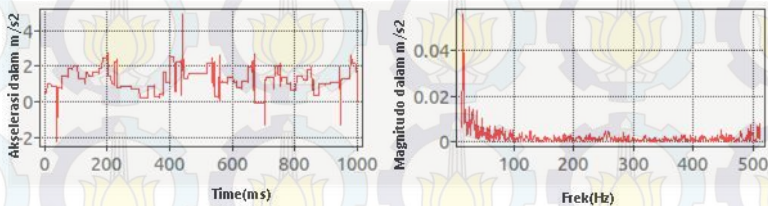
Gambar 4.3 Domain waktu kondisi unfault

4.1.1. Spektrum kondisi *Unfault* (Normal)

Seperti yang telah dijelaskan pada bab 3 di atas bahwa kondisi normal didapatkan saat motor beroperasi normal secara kontinyu. Dari sisi mekanik, dimana kondisi kopling motor dapat diputar dengan mudah secara manual dengan tangan. Pada penelitian ini, diambil beberapa data pada saat motor dinyatakan paa kondisi normal seperti gambar berikut.



Gambar 4.4 Spektrum frekuensi kondisi normal

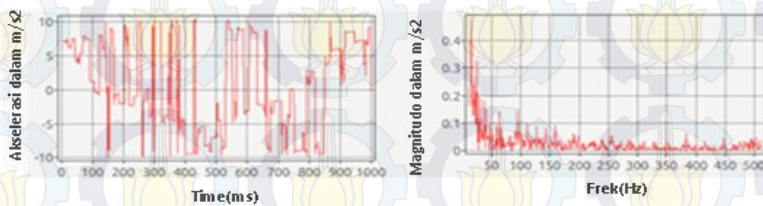


Gambar 4.5 Spektrum frekuensi kondisi normal 2

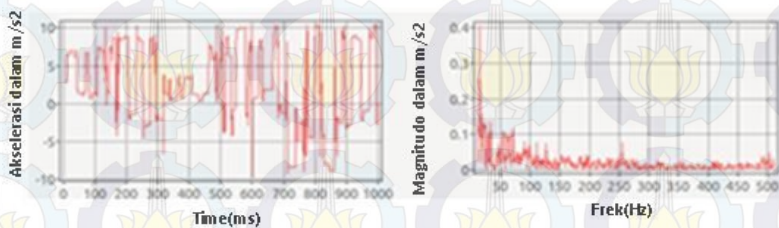
Dari spektrum kondisi normal ini, dapat dilihat bahwa amplitude tertinggi pada pada frekuensi dasar (42,6 Hz). Sementara pada frekuensi-frekuensi harmonik-1,2 dan 3 spektrum tetap muncul dengan amplitude yang lebih kecil.

4.1.2. Spektrum kondisi *Unbalance* (Ketidakseimbangan)

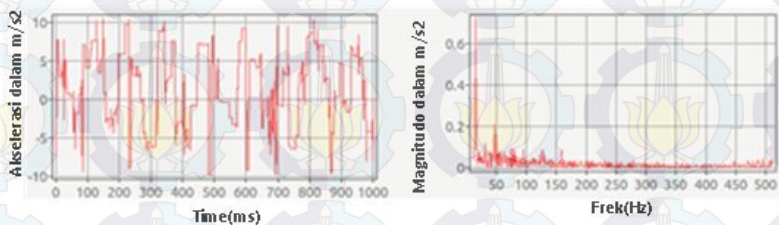
Spektrum frekuensi untuk kondisi *unbalance* terbagi menjadi 4, sesuai dengan ke empat kondisi pengambilan data yang dilakukan pada kondisi unbalance, dimana hasil spektrum vibrasi dan FFT nya pun berbeda, namun perbedaan tersebut masih masuk dalam kondisi unbalance ini. Spektrum frekuensi hasilnya adalah seperti gambar dibawah berikut:



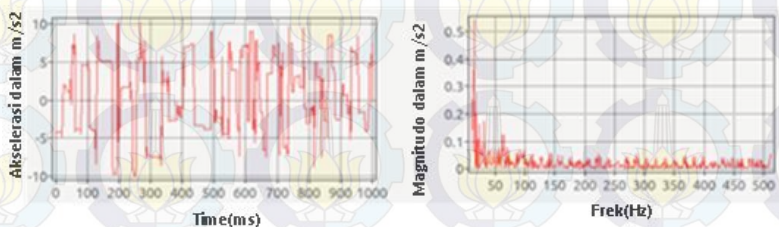
Gambar 4.6 Spektrum frekuensi unbalance kondisi 1



Gambar 4.7 Spektrum frekuensi unbalance kondisi 2



Gambar 4.8 Spektrum frekuensi unbalance kondisi 3



Gambar 4.9 Spektrum frekuensi unbalance kondisi 4

Pada gambar 4.5 dapat dilihat bahwa dengan menaikkan base motor pada kondisi 1 setinggi 2 mm maka akan menyebabkan kondisi unbalance. Hal ini disebabkan karena dengan turunnya satu titik

menyebabkan putaran poros menjadi tidak seimbang. Pada kondisi ini, amplitude pada frekuensi dasar menjadi 2 kali lebih tinggi jika dibandingkan dengan kondisi normal pada frekuensi yang sama. Sementara itu, spektrum pada frekuensi harmoniknya tetap muncul dan nilai amplitudonya tidak berubah hampir sama seperti pada kondisi normal.

Untuk kondisi 2, pola spektrum frekuensi yang akan dihasilkan masih sama dengan spektrum kondisi unbalance. Namun amplitudonya pada frekuensi dasar menurun dan lebih kecil jika dibandingkan pada kondisi 1. Hal ini disebabkan oleh posisi non drive-end motor (kiri ke kanan) mempunyai level lebih tinggi yang sama. Dan hanya berbeda dengan kondisi sisi drive-end motor.

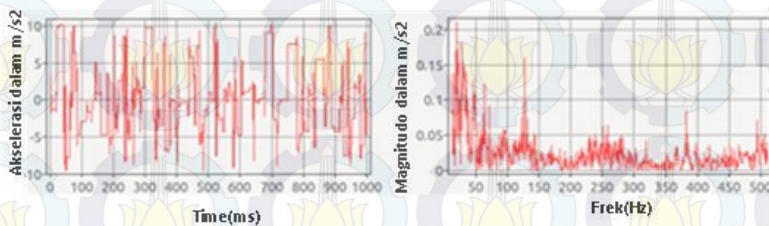
Untuk kondisi 3, amplitude pada frekuensi dasar menjadi lebih tinggi kembali mencapai lebih dari dua kali tinggi amplitude jika dibandingkan dengan kondisi normal pada frekuensi yang sama. Hal ini dikarenakan hanya satu titik yang tidak level seperti kondisi 1 sehingga menyebabkan ketidakseimbangan yang tinggi.

Pada spektrum kondisi 4, amplitude yang dihasilkan pada frekuensi dasar sama seperti dengan kondisi 3. Hal ini dikarenakan titik no.4 masih belum sama levelnya dengan ketiga titik lainnya.

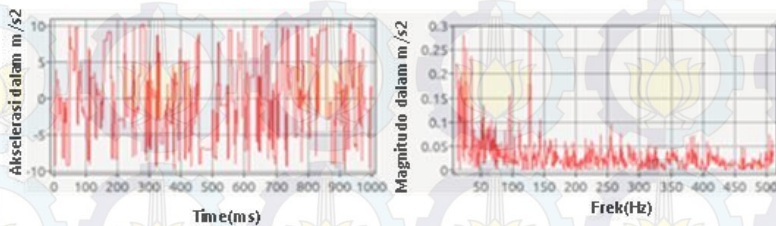
Dari analisis ke-empat spektrum frekuensi untuk kondisi unbalance, dapat disimpulkan bahwa amplitude pada frekuensi dasar menjadi hampir 3 kali besarnya jika dibandingkan dengan kondisi normal. Sementara amplitudonya pada frekuensi harmonik tetap dan hampir sama dengan kondisi normal.

4.1.3. Spektrum kondisi *Misalignment* (Tidak selurus)

Hasil spektrum frekuensi kondisi misalignment terlihat seperti gambar 4.9 dimana dapat dilihat bahwa pada saat misalignment spektrum pada frekuensi harmonik ke 2 mempunyai besar amplitude 2 kali dari amplitude pada kondisi normal. Sementara amplitude pada frekuensi lainnya tetap muncul dan besarnya sama dengan frekuensi dasar.



Gambar 4.10 Spektrum frekuensi misalignment 1



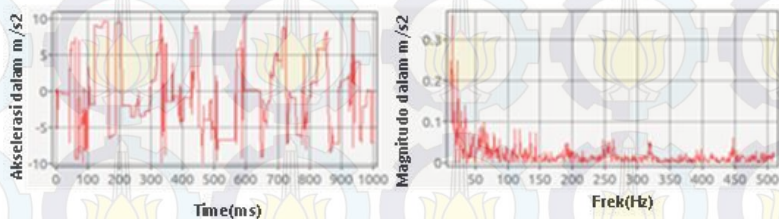
Gambar 4.11 Spektrum frekuensi misalignment 2

Hal ini sesuai dengan penelitian yang telah dilakukan oleh Giovanni Betta (IEEE, 2002) yang menyatakan bahwa kondisi misalignment akan menyebabkan amplitude pada frekuensi harmonic akan dominan. Pada penelitian ini frekuensi harmonic ke 2 yang muncul dominan, kadang lebih tinggi dari frekuensi dasar kadang juga tidak.

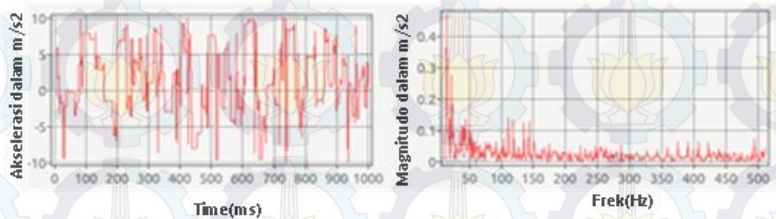
4.1.4. Spektrum kondisi *Looseness* (Kendur/longgar)

Dari gambar dibawah ini dapat dilihat bahwa pada kondisi looseness ini, frekuensi dasar mempunyai amplitude lebih tinggi jika dibandingkan dengan kondisi normal. Hal ini disebabkan karena dengan dikendurkannya semua baut pada base montor menyebabkan ketidakseimbangan menjadi lebih tinggi.

Disamping itu, dari gambar dibawah ini juga dapat dilihat bahwa muncul frekuensi harmonic lainnya pada frekuensi 112 Hz dan 144 Hz. Frekuensi frekuensi ini muncul sebagai akibat kendurnya baut baut pada base motor dan pompa.



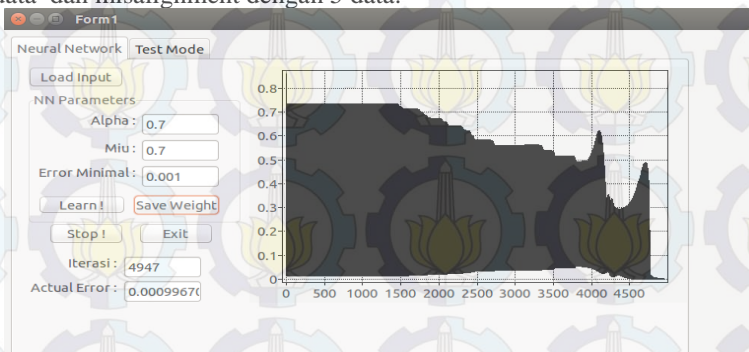
Gambar 4.12 Spektrum frekuensi looseness kondisi 1



Gambar 4.13 Spektrum frekuensi looseness kondisi 2

4.2. Klasifikasi menggunakan neural network

Untuk proses pembelajaran dilakukan dengan menggunakan lazarus di komputer. Dalam proses ini, untuk kondisi normal menggunakan 5 data, unbalance dengan 5 data, looseness dengan 10 data dan misalignment dengan 5 data.



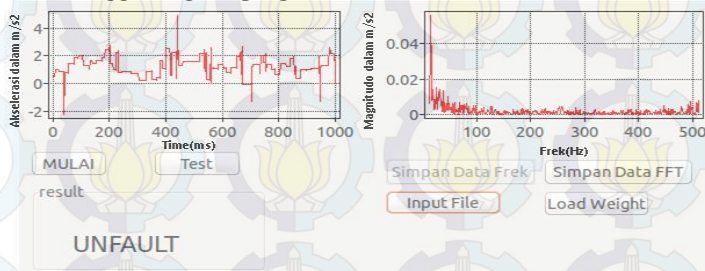
Gambar 4.14 Proses pembelajaran NN di laptop

Dari gambar diatas terlihat bahwa dengan memasukan neural network parameter α dan μ sebesar 0,7 dan menginputkan error minimal yang diinginkan adalah sebesar 0,001. Setelah proses training atau pembelajaran didapatkan bobot yang diinginkan dengan error yang diinginkan setelah melakukan 4947 literasi.

Berikut ini adalah hasil pengujian neural network dengan data yang sudah tersimpan sebelumnya.

Gambar 4.15 Hasil neural network kondisi unfault

Terlihat dalam program Neural Network diatas pada saat kita menginputkan data pada kondisi unfault mempunyai error sekitar 1,5023234 namun program neural network masih mampu mengenalinya sebagai kondisi unfault. Berikut ini adalah tampilan program yang setelah menggabungkan program neural network.



Gambar 4.16 Spektrum vibrasi kondisi unfault

Form1

Neural Network Test Mode

Load Weight

NN's Param

Alpha: 0.7

Miu: 0.7

Error Min: 0.001

Legenda

00 = Looseness

01 = Misalignment

10 = Unbalance

11 = Unfault

NN's Input

Load Input

input[1] = 0.00648

input[2] = 0.0384

input[3] = 0.0378

input[4] = 0.00631

input[5] = 0.00883

input[6] = 0.00198

input[7] = 0.00544

input[8] = 0.0133

input[9] = 0.048

input[10] = 0.0152

input[11] = 0.032

NN's Out

Out[0] = 0.999999924027721

Out[1] = 0.0017363477809200

Error = 1.50745181103888E-6

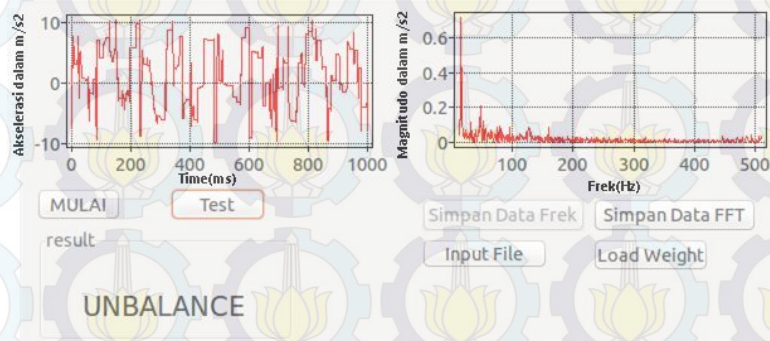
Test!

result

UNBALANCE

Gambar 4.17 Hasil neural network kondisi unbalance

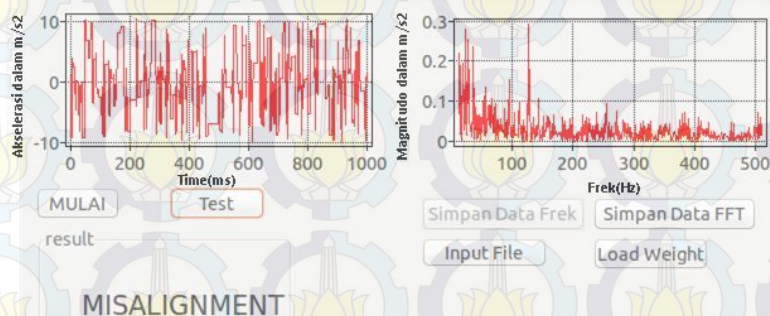
Terlihat dalam program Neural Network diatas pada saat menginputkan data pada kondisi unfault mempunyai error sekitar 1,5074 namun program neural network masih mampu mengenalinya sebagai kondisi unbalance. Berikut ini adalah tampilan program yang setelah menggabungkan program neural network.



Gambar 4.18 Spektrum vibrasi kondisi unbalance

Gambar 4.19 Hasil neural network kondisi misalignment

Terlihat dalam program Neural Network diatas pada saat menginputkan data pada kondisi unfault mempunyai error sekitar 0,00315 namun program neural network masih mampu mengenalinya sebagai kondisi misalignment. Berikut ini adalah tampilan program yang setelah menggabungkan program neural network.



Gambar 4.20 Spektrum vibrasi kondisi misalignment

Form1

Neural Network Test Mode

Load Weight

NN's Param

Alpha : 0.7

Miu : 0.7

Error Min : 0.001

Legenda

00 = Looseness

01 = Misalignment

10 = Unbalance

11 = Unfault

NN's Input

Load Input

input[1] = 0.00625

input[2] = 0.0305

input[3] = 0.00898

input[4] = 0.00321

input[5] = 0.0123

input[6] = 0.0138

input[7] = 0.00381

input[8] = 0.00168

input[9] = 0.0225

input[10] = 0.00263

input[11] = 0.0171

NN's Out

Out[0] = 0.0001536605423762

Out[1] = 0.00845874027540298

Error = 0.00003578694930450

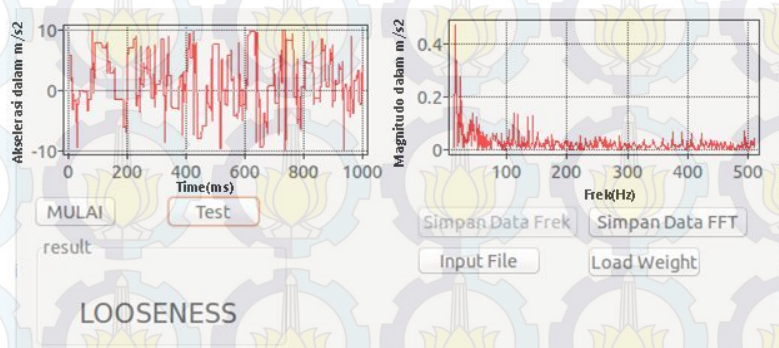
Test !

result

LOOSENESS

Gambar 4.21 Hasil neural network kondisi looseness

Terlihat dalam program Neural Network diatas pada saat menginputkan data pada kondisi unfault mempunyai error sekitar 0,00003 namun program neural network masih mampu mengenalinya sebagai kondisi looseness. Berikut ini adalah tampilan program yang setelah menggabungkan program neural network.



Gambar 4.22 Spektrum vibrasi kondisi looseness

Tabel 4.1 Hasil Pengujian NN

No.	Pola yang muncul	Hasil Neural Network	Hasil
1	Unfault	Unfault	Tepat
2	Unfault	Unfault	Tepat
3	Unbalance	Unbalance	Tepat
4	Looseness	Misalignment	Tidak Tepat
5	Misalignment	Misalignment	Tepat
6	Looseness	Looseness	Tepat
7	Unfault	Unfault	Tepat
8	Misalignment	Looseness	Tidak tepat
9	Unfault	Unfault	Tepat
10	Unbalance	Unbalance	Tepat

Dari tabel 4.1, dapat dilihat bahwa tingkat keberhasilan neural network adalah 80 %.

DAFTAR PUSTAKA

- [1] Ahmed, R. Supangat, J. Grieger, N. Ertugrul and W. L. Soong, (2004), "A Baseline Study for On-Line Condition Monitoring of Induction Machines", Australasian Universities Power Engineering Conference (AUPEC 2004).
- [2] Birajdar R, Pati R, Khanzode K, Kirloskar Brothers Ltd., (2009), "*Vibration And Noise in Centrifugal Pumps - Sources And Diagnosis Methods*", 3rd International Conference on Integrity, Reliability and Failure, Portugal, 20-24 July 2009..
- [3] Brian T. Holm-Hansen, Robert X. Gao, (2000), "Vibration Analysis of a Sensor Integrated Ball Bearing", Vol. 122, Hal 384 ~ 392, Transactions of the ASME.
- [4] Giovanni Betta, Student Member, IEEE, Consolatina Liguori, Alfredo Paolillo, and Antonio Pietrosanto, (2002), "A Dsp-Based Fft-Analyzer For The Fault Diagnosis Of Rotating Machine Based On Vibration Analysis", IEEE Transactions on Instrumentation and Measurement, vol. 51, no. 6.
- [5] H. Shreve Dennis, (1995), "Signal Processing For Effective Vibration Analysis", IRD Mechanalysis, Ins Columbus, Ohio.
- [6] Reimche W, Südmersen U, Pietsch Or, Scheer C, Bach F, (2003), "*Basics Of Vibration Monitoring For Fault Detection And Process Control*". University of Hannover, Institute of Material Science, Department of NDT.
- [7] Ruuska, Mikko, H. Andersson, Paul, Dubrovnik, (2003), "*Spindle Bearing Monitoring Using Acoustic Emission*", Proceedings, XVII IMEKO World Congress, 2164 ~ 2167.
- [8] Stephan J. Chapman. *Electric Machinery Fundamentals* (4th ed), New York: McGraw Hill, Higher Education, 2005.
- [9] Moh. Ishak, Analisis Proteksi Vibrasi Pada Pompa Sentrifugal Dengan Menggunakan Fast Fourier Transform Dan Neural Network, 2009.



Halaman ini sengaja dikosongkan

BAB 5

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Kondisi unjuk kerja motor dapat diamati dengan menggunakan spektrum vibrasi. Kondisi unjuk kerja motor dapat diamati dengan menggunakan spektrum vibrasi. Pada kondisi normal dengan beban dan tanpa beban terjadi perbedaan tinggi amplitudo. Pada kondisi unbalance, spektrum frekuensi dasar menjadi 3 kali lebih besar dari kondisi normal. Pada kondisi misalignment, spektrum frekuensi harmonik ke-2 akan lebih dominan. Pada kondisi looseness, spektrum frekuensi dasar akan naik menjadi 2 kali lebih tinggi dari kondisi normal. Tingkat Keberhasilan neural network mencapai 80%. Dengan menggunakan sistem online spektrum vubrasi berbasis FFT dan neural network pada motor akan dapat mengetahui lebih awal kondisi kondisi alignment, unbalance, dan looseness.

5.2. Saran

Proses alignment antara pompa dan motor pada kondisi awal harus lebih presisi, karena menentukan kondisi unfault motor. Proses monitoring akan lebih mudah dan efisien jika menggunakan monitor touchscreen. Penggunaan sensor accelerometer yang standart industri lebih disarankan agar data hasil vibrasinya lebih baik. Pembuatan peletakan sensor yang kuat dan menempel dengan motor akan menghasilkan pengambilan data yang lebih baik.



Halaman ini sengaja dikosongkan

LAMPIRAN

Listing program di arduino:

```
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_ADXL345_U.h>

/* Assign a unique ID to this sensor at the same time */
Adafruit_ADXL345_Unified accel =
Adafruit_ADXL345_Unified(12345);
void outBit(float Datin1)// merubah float jadi biner
{
  //digitalWrite(2, Datin1%2);
  Datin1 = (Datin1+10)*50;
  int Datin = Datin1;
  Serial.print(Datin); Serial.print(" ");
  for(int i=2;i<12;i++) //Datin>1
  {
    Serial.print(Datin%2);
    digitalWrite(i, Datin%2); // yang dikirim ke raspberry
    Datin = Datin/2;
  }
  Serial.println("");
  return;
}

void setup(void)
{ // initialize digital pin 13 as an output.
  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);
  pinMode(8, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(10, OUTPUT);
  pinMode(11, OUTPUT);
  Serial.begin(9600);
```



```

/* Initialise the sensor */
if(!accel.begin())
{
    /* There was a problem detecting the ADXL345 ... check your
connections */
    Serial.println("Ooops, no ADXL345 detected ... Check your
wiring!");
    while(1);
}
void loop(void)
{
    /* Get a new sensor event */
    sensors_event_t event;
    accel.getEvent(&event);
    outBit(event.acceleration.y);
    //delay(8);
}

```

Listing program di lazarus raspberry pi:

```

unit Unit1;

{$mode objfpc} {$H+}

interface


uses
    Classes, SysUtils, FileUtil, Forms, Controls, Graphics, Dialogs,
    ComCtrls,
    StdCtrls, ExtCtrls, TAGraph, TASeries, rpi_hal, unix, BaseUnix;

type

    { TForm1 }

    TForm1 = class(TForm)
        Button1: TButton;
        Button2: TButton;
        Button4: TButton;

```



```

Button5: TButton;
Button6: TButton;
ButtonInputFile: TButton;
ButtonMulai: TButton;
Chart1: TChart;
Chart1LineSeries1: TLineSeries;
Chart1LineSeries2: TLineSeries;
Chart1LineSeries3: TLineSeries;
Chart1LineSeries4: TLineSeries;
Chart2: TChart;
Chart2LineSeries1: TLineSeries;
Chart2LineSeries2: TLineSeries;
Chart2LineSeries3: TLineSeries;
Chart2LineSeries4: TLineSeries;
Edit1: TEdit;
Edit2: TEdit;
GroupBox6: TGroupBox;
Label16: TLabel;
OpenDialog1: TOpenDialog;
OpenDialog2: TOpenDialog;
OpenDialog3: TOpenDialog;
SaveDialog1: TSaveDialog;
SaveDialog2: TSaveDialog;
Timer1: TTimer;
procedure Button1Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure Button4Click(Sender: TObject);
procedure Button5Click(Sender: TObject);
procedure Button6Click(Sender: TObject);
procedure ButtonInputFileClick(Sender: TObject);
procedure ButtonMulaiClick(Sender: TObject);
procedure ButtonSettingClick(Sender: TObject);
procedure ComboBoxJumlahDataChange(Sender: TObject);
procedure ComboBoxNilaiMaksimumChange(Sender: TObject);
procedure FormClose(Sender: TObject; var CloseAction:
TCloseAction);
procedure FormCreate(Sender: TObject);
procedure PageControl1Change(Sender: TObject);

```

```

procedure ComPort1RxChar();
procedure load_weight();
procedure load_input();
procedure feedforward();
procedure backpropagation();
private
{ private declarations }
public
{ public declarations }
end;

var
Form1: TForm1;
accel_data: array[1..4,1..10000] of extended;
data_frekuensi: array[1..10000] of extended;
sst_state,sst_data_length: integer;
rx_index: byte;
rx_buff: array[1..250] of byte;
str, nilai, sumbu: string;
com_state: boolean;
cksum,temp: byte;
data_index,data_buffer: integer;
xasalr,xasali:array[1..10000] of extended;
N,index,panjang_fft:integer;
x1,xi,xr,phi,power,w:array[1..10000] of extended;
filename:textfile;
indexfft: integer;
amp: array[1..4,1..10000]of extended;
nambah : integer;

z: integer;
iterasi: integer;
miu,alpha,erormin,eror: extended;
temp2: Textfile;
tombol: integer;
buff: string;
erkum: array[1..25] of extended;
input: array[1..26,1..500] of extended;

```



```

node1,turunan1,out1,error1: array[1..60] of extended;
node2,turunan2,out2,error2: array[1..20] of extended;
node3,turunan3,out3,error3: array[1..2] of extended;
w1: array[1..500,1..60] of extended;
w2: array[1..60,1..20] of extended;
w3: array[1..20,1..2] of extended;

const
    desired_output: array[1..25,1..2] of integer=
        ((0,0),(0,0),(0,0),(0,0),(0,0),
          (0,0),(0,0),(0,0),(0,0),(0,0),
          (0,1),(0,1),(0,1),(0,1),(0,1),
          (1,0),(1,0),(1,0),(1,0),(1,0),
          (1,1),(1,1),(1,1),(1,1),(1,1));

```

implementation

```
{SR *.lfm}
```

```

procedure TForm1.ButtonSettingClick(Sender: TObject);
begin
    //ComPort1.ShowSetupDialog;
end;

```

```

procedure TForm1.PageControl1Change(Sender: TObject);
begin
end;

```

```

// Program FFT //
procedure fft(t:extended);
label satu,dua,tiga,empat;
var
    lim1,lim2,lim3,l,r,m,k,i,j:integer;
    d,x1,x2,b1,b2,c1,c2,arg,sin1,cos1:extended;
begin
    m:= N;
    for k:= 1 to panjang_fft do

```

```

begin
  xr[k]:=xasalr[k];
  xi[k]:=xasali[k];
end;

if t=1.0 then
  d:=1.0
else
  d:=panjang_fft;

  {suffle data input}
  lim1:=panjang_fft-1;
  lim2:= round(panjang_fft/2);

  j:= 1;
  for i:= 1 to lim1 do
    begin
      if i > j-1 then
        goto dua;
      x1:=xr[j];
      x2:=xi[j];
      xr[j]:=xr[i];
      xi[j]:=xi[i];
      xr[i]:=x1;
      xi[i]:=x2;
    dua:
      l:=lim2;
    tiga:
      if l > j-1 then goto empat;
      j:=j-l;
      l:=round(l/2);
      goto tiga;
    empat:
      j:=j+l;
    end;
  }
  {inplace transformation}
  for i:= 1 to m do
    begin

```

```

lim1:= round(exp((i-1)*ln(2)));
lim2:= round(exp((m-i)*ln(2)));
for l:= 1 to lim2 do
begin
  for r:= 1 to lim1 do
  begin
    lim3:= (r-1)+(l-1)*2*lim1+1;
    b1:=xr[lim3];
    b2:=xi[lim3];
    c1:=xr[lim3+lim1];
    c2:=xi[lim3+lim1];
    arg:=2*pi*(r-1)*lim2/panjang_fft;
    cos1:=cos(arg);
    sin1:=sin(arg);
    x1:=(c1*cos1)+(c2*sin1*t);
    x2:=(-c1*sin1*t)+(c2*cos1);
    xr[lim3]:=b1+x1;
    xi[lim3]:=b2+x2;

    xr[lim3+lim1]:=b1-x1;
    xi[lim3+lim1]:=b2-x2;
  end;
end;
end;
for k:= 1 to panjang_fft do
begin
  xr[k]:=xr[k] / d;
  xi[k]:=xi[k] / d;
end;
end;

// Biner To Integer //
function BinToInt(value: string): Integer;
var
  i, iValueSize: Integer;
begin
  Result := 0;

```



```

iValueSize := Length(value);
for i:= iValueSize downto 1 do
if Value[i] = '1' then Result := Result + (1 shl (iValueSize - i));
end;

```

```

procedure TForm1.ComPort1RxChar();
var
str: string;
sst_state,sst_data_length,chk: byte;
j,k,l,m: integer;
i: byte;
begin
data_index:=1;
Chart1LineSeries1.Clear;
while data_index < 1000 do
begin
nilai:="";
sumbu:="";

for l:=11 downto 2 do
begin
rpi_hal.gpio_set_input(l);
if rpi_hal.gpio_get_PIN(l) then nilai:= nilai + '1'
else nilai:= nilai + '0';
end;

```

```

Edit2.Text:=nilai;
accel_data[1][data_index]:=(BinToInt(nilai) / 50)-10;

```

```

Edit1.Text:=FloatToStr(BinToInt(nilai));

```

```
Chart1LineSeries1.AddXY(data_index,(accel_data[1][data_index]));
```

```
data_index:=data_index+1;
```

```
//sleep(9); //delay 1 milisecond
```

```
//Application.ProcessMessages;
```

```
if data_index=1000 then
```

```
begin
```

```
for j :=1 to 1 do
```

```
begin
```

```
for k:= 1 to data_index do //1024 do
```

```
begin
```

```
xasalr[k]:=accel_data[j][k]; //data_frekuensi[i];
```

```
xasali[k]:=0.0;
```

```
end;
```

```
N:=round(ln(data_index)/ln(2));
```

```
panjang_fft:=1 shl N ;
```

```
fft(0.0);
```

```
indexfft:=panjang_fft div 2;
```

```
for k:= indexfft+1 to panjang_fft do
```

```
begin
```

```
amp[j][k-indexfft]:= sqrt(xr[k]*xr[k]+xi[k]*xi[k]);
```

```
end;
```

```
for k:= 1 to panjang_fft div 2 do
```

```
begin
```

```
amp[j][k+indexfft]:= sqrt(xr[k]*xr[k]+xi[k]*xi[k]);
```

```
end;
```

```
Chart2LineSeries1.Clear;
```

```
for k:= 1 to round(data_index/2) do
```

```
begin
```

```
Chart2LineSeries1.AddXY(indexfft-k,amp[j][k]);
```

```
end;
```

```
Button1.Enabled:=true;
```

```
Button2.Enabled:=true;
```

```
panjang_fft:=data_index;
```

```

        end;
        // data_index:=1;
    end;

end;
for n:= 1 to 500 do
begin
    input[26,n]:= amp[1][n];
end;
end;

procedure TForm1.FormCreate(Sender: TObject);
begin
    nambah := 0;
    Button1.Enabled:=false;
    Button2.Enabled:=false;
    data_index:=1;
end;

procedure TForm1.ButtonMulaiClick(Sender: TObject);
begin
    if ButtonMulai.Caption='MULAI' then
    begin
        ButtonMulai.Caption:='BERHENTI';
        ComPort1RxChar();
    end
    else
    begin
        cksum:=0;
        cksum:=$1E+$28+$01+$00+$01+$01+$00;
        cksum:=0-cksum;
        ButtonMulai.Caption:='MULAI';
        rx_index:=1;
    end;
end;

```



```

end;

procedure TForm1.ComboBoxJumlahDataChange(Sender: TObject);
begin
end;

procedure TForm1.FormClose(Sender: TObject; var CloseAction:
TCloseAction);
begin
end;

procedure TForm1.ComboBoxNilaiMaksimumChange(Sender:
TObject);
begin
end;

procedure TForm1.ButtonInputFileClick(Sender: TObject);
var
jumlah_data,i: integer;
number: array[1..10000] of integer;
j: extended;
begin
  if OpenFileDialog1.Execute then
  begin
    AssignFile(filename,OpenDialog1.FileName);
    reset(filename);
    jumlah_data:=1;
    while not(eof(filename)) do
    begin
      readln(filename,data_frekuensi[jumlah_data]);
      inc(jumlah_data);
    end;
    jumlah_data:=jumlah_data-1;
    j:=jumlah_data/20+10;
    closefile (filename);
    Chart1LineSeries1.Clear;
  end;
end;

```

```

for i:= 1 to jumlah_data do //1024 do
begin
  j:=i/20;
  Chart1LineSeries1.AddXY(i,data_frekuensi[i]);
end;
for i:= 1 to jumlah_data do //1024 do
begin
  xasalr[i]:=data_frekuensi[i];
  xasali[i]:=0.0;
end;
N:=round(ln(jumlah_data)/ln(2));
panjang_fft:=1 shl N ;
fft(0.0);
indexfft:=panjang_fft div 2;
for i:= indexfft+1 to panjang_fft do
begin
  amp[1][i-indexfft]:= sqrt(xr[i]*xr[i]+xi[i]*xi[i]);
end;
for i:= 1 to panjang_fft div 2 do
begin
  amp[1][i+indexfft]:= sqrt(xr[i]*xr[i]+xi[i]*xi[i]);
end;
Chart2LineSeries1.Clear;
for i:= 1 to round(jumlah_data/2) do
begin
  Chart2LineSeries1.AddXY(indexfft-i,amp[1][i]);
end;
Button2.Enabled:=true;
panjang_fft:=jumlah_data;

end;
for i:= 1 to 500 do
begin
  input[26,i]:= amp[1][i];
end;
end;

procedure TForm1.Button2Click(Sender: TObject);

```

```

var
i: integer;
begin
  if SaveDialog1.Execute then
  begin
    SaveDialog1.FileName:=SaveDialog1.FileName+'.txt';
    AssignFile(filename,SaveDialog1.FileName);
    rewrite(filename);
    for i:= 1 to round(panjang_fft/2) do
    begin
      writeln(filename,#$09,amp[1][i]);//,$$09,amp[2][i],

    end;
    closefile(filename);
  end;
end;

procedure TForm1.Button3Click(Sender: TObject);
begin
end;

procedure TForm1.Button4Click(Sender: TObject);
var
l : integer;
begin
  nambah := nambah + 1;
  nilai:="";
  sumbu:="";

  for l:=8 downto 1 do
  begin
    rpi_hal.gpio_set_input(l);
    if rpi_hal.gpio_get_PIN(l) then nilai:= nilai +'1'
    else nilai:= nilai +'0';
  end;

```



```

Edit1.Text:=IntToStr((BinToInt(nilai) div 10)-10);
Chart1.LineSeries1.AddXY(nambah,((BinToInt(nilai) div 10)-10));
end;

```

```

procedure TForm1.Button1Click(Sender: TObject);
var
i: integer;
begin
if SaveDialog1.Execute then
begin
SaveDialog1.FileName:=SaveDialog1.FileName+'.txt';
AssignFile(filename,SaveDialog1.FileName);
rewrite(filename);
ShowMessage(IntToStr(data_index));
for i:= 1 to data_index do
begin

```

```

writeln(filename,#$09,accel_data[1][i]);//,$$09,accel_data[2][i],#$09,
end;
closefile(filename);
end;
end;

```

```

procedure tform1.load_weight;
var
i,j: integer;
begin
if OpenFileDialog1.Execute then
begin
assignfile(temp2,opendialog1.FileName);
reset(temp2);
for i:= 1 to 500 do
begin
for j:= 1 to 60 do
begin
readln(temp2,buff);
w1[i,j]:= strtofloat(buff);

```

```

    end;
end;

for i:= 1 to 60 do
begin
    for j:= 1 to 20 do
    begin
        readln(temp2,buff);
        w2[i,j]:= strtofloat(buff);
    end;
end;

for i:= 1 to 20 do
begin
    for j:= 1 to 2 do
    begin
        readln(temp2,buff);
        w3[i,j]:= strtofloat(buff);
    end;
end;
readln(temp2,buff);
alpha:= strtofloat(buff);

readln(temp2,buff);
miu:= strtofloat(buff);

readln(temp2,buff);
erormin:= strtofloat(buff);

closefile(temp2);
end;
end;

procedure TForm1.load_input;
begin
end;

```

```
procedure TForm1.Button5Click(Sender: TObject);  
begin  
    load_weight;  
end;
```

```
procedure TForm1.Button6Click(Sender: TObject);  
var  
    x,y: integer;  
    init_eror: extended;  
    pos: integer;  
begin  
    z:= 26;  
    feedforward;
```

```
    init_eror:= 1;
```

```
    for x:= 1 to 25 do  
    begin  
        erkum[x]:=0;  
        for y:=1 to 2 do  
            begin
```

```
                erkum[x]:= erkum[x]+sqr(desired_output[(x),y]-out3[y]);
```

```
            end;
```

```
            erkum[x]:= 0.5*(erkum[x]);
```

```
            if erkum[x]< init_eror then
```

```
                begin
```

```
                    init_eror:=erkum[x];
```

```
                    pos:= x;
```

```
                end;
```

```
            end;
```

```
            if (pos<=10) then
```

```
                begin
```

```
                    label16.Caption:= 'LOOSENESS';
```

```
                end
```

```
            else if (pos>10) and (pos<= 15) then
```

```
                begin
```

```
                    label16.Caption:= 'MISALIGNMENT';
```



```

end
else if (pos>15) and (pos<= 20) then
begin
    label16.Caption:= 'UNBALANCE';
end
else if (pos>20) and (pos<= 25) then
begin
    label16.Caption:= 'UNFAULT';
end
else
begin
    label16.Caption:= 'UNKNOWN';
end;
end;

procedure tform1.feedforward;
var
    i,j: integer;
begin
    for i:= 1 to 60 do
        begin
            node1[i]:= 0;
            for j:= 1 to 500 do
                begin
                    node1[i]:= node1[i]+(w1[j,i]*input[z,j]);
                end;
            out1[i]:= 1/(1+exp(-1*alpha*node1[i]));
            turunan1[i]:= alpha*out1[i]*(1-out1[i]);
        end;

        for i:= 1 to 20 do
            begin
                node2[i]:= 0;
                for j:= 1 to 60 do
                    begin
                        node2[i]:= node2[i]+(w2[j,i]*out1[j]);
                    end;
                out2[i]:= 1/(1+exp(-1*alpha*node2[i]));
                turunan2[i]:= alpha*out2[i]*(1-out2[i]);
            end;
        end;
    end;
end;

```

```

end;

for i:= 1 to 2 do
begin
    node3[i]:= 0;
    for j:= 1 to 20 do
        begin
            node3[i]:= node3[i]+(w3[j,i]*out2[j]);
        end;
    out3[i]:= 1/(1+exp(-1*alpha*node3[i]));
    turunan3[i]:= alpha*out3[i]*(1-out3[i]);
end;
end;

procedure tform1.backpropagation;
var
    i,j: integer;
begin
    for i:= 1 to 2 do //error output (layer 3)
        begin
            eror3[i]:= (desired_output[z,i]-out3[i])*turunan3[i];
        end;
        erkum[z]:= 0;

        for i:= 1 to 2 do
            begin
                erkum[z]:= erkum[z]+sqr(desired_output[z,i]-out3[i]);
            end;

            erkum[z]:= (0.5)*erkum[z];

            for i:=1 to 20 do
                begin
                    eror2[i]:= 0;
                    for j:= 1 to 2 do
                        begin
                            eror2[i]:= eror2[i]+(eror3[j]*w3[i,j]);

```




```
end;  
eror2[i]:= eror2[i]*turunan2[i]; //error layer 2  
end;
```

```
for i:=1 to 60 do  
begin  
  eror1[i]:= 0;  
  for j:= 1 to 20 do  
  begin  
    eror1[i]:= eror1[i]+(eror2[j]*w2[i,j]);  
  end;  
  eror1[i]:= eror1[i]*turunan1[i]; //error layer 1  
end;  
end;
```

```
end.
```

```
end.
```

Halaman ini sengaja dikosongkan



ITS
Institut
Teknologi
Sepuluh Nopember

FINAL PROJECT - TE 141599

**ABNORMALITIES CLUTCH DETECTION IN INDUCTION
MOTOR USING ACCELEROMETER SENSOR BASED ON
RASPBERRY PI**

Rahmadi Ridho Anjiko
NRP 2211 100 142

Advisor
Dr. Muhammad Rivai, ST., MT.
Suwito, ST., MT.

ELECTRICAL ENGINEERING DEPARTEMENT
Faculty of Industrial Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2015

DETEKSI KETIDAKNORMALAN KOPLING PADA MOTOR INDUKSI MENGGUNAKAN SENSOR ACCELEROMETER BERBASIS RASPBERRY PI

Nama : Rahmadi Ridho Anjiko
NRP : 2211100142
Jurusan : Teknik Elektro
Dosen Pembimbing : 1. Dr. Muhammad Rivai. ST., MT.
2. Suwito, ST., MT.

ABSTRAK

Motor induksi sangat banyak digunakan dalam kehidupan sehari-hari kita. Baik itu digunakan di industri maupun di rumah tangga karena konstruksinya sederhana, pengoperasian mudah dan mempunyai kecepatan yang relative konstan. Pemeliharaan/perawatan mesin sangat dibutuhkan untuk memperpanjang umur pemakaian motor induksi. Tindakan preventif harus didahulukan untuk mencegah kerusakan yang lebih besar pada mesin. Salah satu tindakan preventif yang dapat dilakukan adalah monitoring vibrasi pada mesin.

Kerusakan pada motor induksi yang tidak terdeteksi pada tahap awal dapat menyebabkan kerusakan lebih parah. Kerusakan motor di industri dapat mengakibatkan kerugian yang sangat besar karena proses produksi berhenti. Dalam penelitian ini, digunakan sensor accelerometer untuk melihat pola vibrasi sinyal dengan menggunakan Fast Fourier Transform (FFT) dan Neural Network terhadap jenis-jenis kondisi abnormal. Ada tiga jenis kondisi abnormal yang dibahas dalam penelitian ini yaitu *misalignment*, *unbalance* dan *looseness*. Memonitoring keadaan mesin listrik secara kontinyu sehingga keadaan abnormal pada motor listrik dapat diketahui secara dini.

Sistem dapat mendeteksi jenis-jenis kerusakan secara on-line pada ketidaknormalan kopling motor listrik yaitu unbalance, misalignment dan looseness. Tingkat keberhasilan neural network dalam sistem ini mencapai 80 persen.

Kata Kunci: *Fast fourier transform, Motor induksi, Neural Network, Sensor accelerometer, Vibrasi.*



Halaman ini sengaja dikosongkan

ABNORMALITIES CLUTCH DETECTION IN INDUCTION MOTOR USING ACCELEROMETER SENSOR BASED ON RASPBERRY PI

Name : Rahmadi Ridho Anjiko
NRP : 2211100142
Department : Electrical Engineering
Adivsor : 1. Dr. Muhammad Rivai, ST., MT.
2. Suwito, ST., MT.

ABSTRACT

Induction motors are very widely used in our daily lives. Whether it is used in industry and in households because their construction is simple, easy operation and has a relatively constant speed. Engine maintenance is needed to extend the life of the induction motor. Preventive measures should be prioritized in order to prevent greater damage to the machine. One of the preventive measures that can be done is monitoring vibrations in the engine.

Damage in the induction motor is not detected at an early stage can cause more severe damage. Damage to the motor in the industry can result in huge losses due to the production process stops. In this study, uses accelerometer sensor to see the pattern of vibration signals using Fast Fourier Transform (FFT) and Neural Network on the types of abnormal conditions. There are three types of abnormal conditions that are discussed in this study are misalignment, unbalance and looseness. Monitoring the state of the electric machine continuously so that the abnormal state of the electrical machine can be known at an early stage.

The system can detect the type of damage on the on-line coupling an electric motor abnormalities such as unbalance , misalignment and looseness . The success rate of neural network in the system reached 80 percent.

Keywords: *Fast fourier transform, Induction Motor, Neural Network, Sensor accelerometer, Vibration.*



Halaman ini sengaja dikosongkan

KATA PENGANTAR

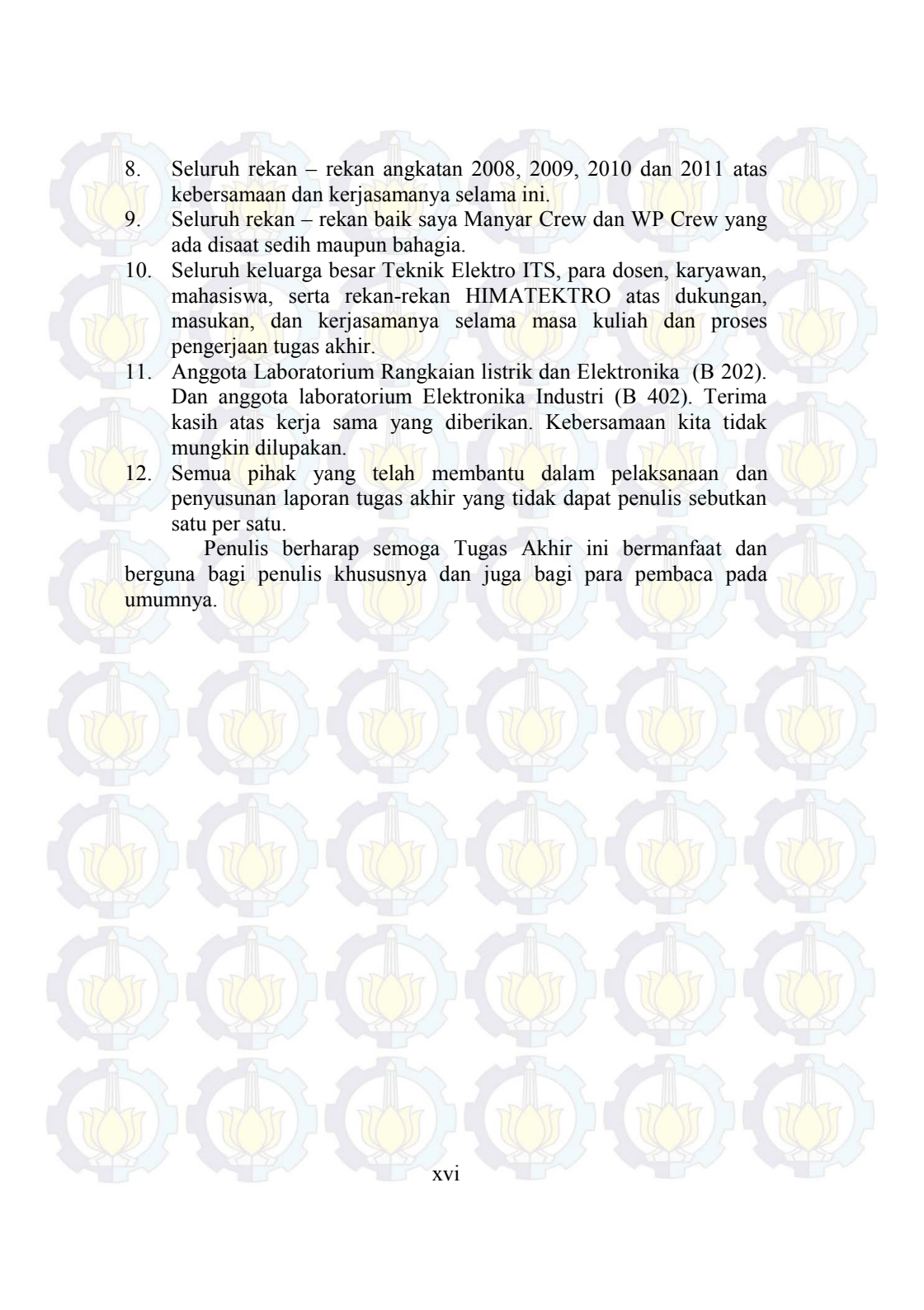
Segala puji syukur bagi Tuhan Yang Maha Esa atas kemudahan, kesehatan, dan kesempatan yang telah dikaruniakan dari awal hingga akhir pengerjaan tugas akhir berjudul:

Deteksi Ketidaknormalan Kopling pada Motor Induksi menggunakan Sensor Accelerometer berbasis Raspberry Pi

Penyusunan Tugas Akhir ini adalah sebagai salah satu persyaratan untuk menyelesaikan jenjang pendidikan S1 pada bidang studi Elektronika di Jurusan Teknik Elektro ITS Surabaya, selain itu juga diharapkan mampu menambah wawasan dalam bidang keahlian elektronika.

Dalam penyusunan Tugas Akhir ini dan selama studi di Institut Teknologi Sepuluh Nopember, penulis mendapatkan bantuan, bimbingan, dan dukungan tak ternilai dari berbagai pihak. Untuk itu penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Bapak Dr. Muhammad Rivai, ST., MT. selaku dosen pembimbing pertama, atas kesediaannya memberikan masukan dan waktu untuk menyelesaikan tugas akhir ini.
2. Bapak Suwito, ST., MT. dosen pembimbing kedua, atas bimbingan, perhatian dan penjelasan serta motivasi yang diberikan.
3. Seluruh Dosen Bidang Studi Elektronika Jurusan Teknik Elektro ITS, atas bimbingan serta ilmu - ilmu perkuliahan yang telah diberikan.
4. Kedua orang tua yang tercinta: Sunaryo dan Siti Samsiyah yang selalu memberikan do'a, nasihat, dukungan serta motivasi yang telah diberikan untuk keberhasilan penulis.
5. Saudara saya yang tercinta: Rizal, Rissa yang membut penulis termotivasi untuk menjadi teladan sejati sebagai adik yang baik.
6. Keluarga Besar E 51 Elektro ITS yang selalu memberi semangat dan selalu membantu memberi motivasi yang sangat luar biasa.
7. Teman – teman seperjuangan :Nafi, Bintang, Kuncoro, Daus, Imam, Paulus, Helmi, dan khususnya Reza dan agung atas waktu, ide – ide, semangat dan dukungan yang diberikan kepada penulis. Terima kasih atas kerja sama yang diberikan. Kebersamaan kita tidak mungkin dilupakan.

- 
8. Seluruh rekan – rekan angkatan 2008, 2009, 2010 dan 2011 atas kebersamaan dan kerjasamanya selama ini.
 9. Seluruh rekan – rekan baik saya Manyar Crew dan WP Crew yang ada disaat sedih maupun bahagia.
 10. Seluruh keluarga besar Teknik Elektro ITS, para dosen, karyawan, mahasiswa, serta rekan-rekan HIMATEKTRO atas dukungan, masukan, dan kerjasamanya selama masa kuliah dan proses pengerjaan tugas akhir.
 11. Anggota Laboratorium Rangkaian listrik dan Elektronika (B 202). Dan anggota laboratorium Elektronika Industri (B 402). Terima kasih atas kerja sama yang diberikan. Kebersamaan kita tidak mungkin dilupakan.
 12. Semua pihak yang telah membantu dalam pelaksanaan dan penyusunan laporan tugas akhir yang tidak dapat penulis sebutkan satu per satu.

Penulis berharap semoga Tugas Akhir ini bermanfaat dan berguna bagi penulis khususnya dan juga bagi para pembaca pada umumnya.

DAFTAR ISI

PERNYATAAN KEASLIAN TUGAS AKHIR	vii
LEMBAR PENGESAHAN	ix
ABSTRAK	xi
ABSTRACT	xiii
KATA PENGANTAR	xv
DAFTAR ISI	xvii
DAFTAR GAMBAR	xxi
DAFTAR TABEL	xxv
BAB 1 PENDAHULUAN	1
1.1. Latar Belakang.....	1
1.2. Perumusan Masalah.....	2
1.3. Batasan Permasalahan	2
1.4. Tujuan Permasalahan.....	3
1.5. Metode Penelitian.....	3
1.6. Sistematika Penulisan.....	4
BAB 2 TINJAUAN PUSTAKA	5
2.1. Analisa Vibrasi	5
2.1.1. Analisa Vibrasi dan Noise pada Motor Induksi – Sumber Vibrasi dan Metode Diagnosanya.....	5
2.1.2. Dasar-dasar Monitor Vibrasi Untuk Mendeteksi Kegagalan dan Control Proses	5
2.1.3. Analizer FFT Untuk Diagnosa Kegagalan pada Mesin Berputar Berdasarkan Analisa Vibrasi.....	6
2.2. Motor Induksi	6
2.3. Getaran	7

2.3.1.	Penyebab Vibrasi.....	9
2.3.2.	Accelerometer MEMS (Micro-Electro Mechanical System).....	10
2.3.3.	Accelerometer ADXL345 MEMS.....	12
2.4.	Raspberry Pi.....	13
2.5.	Arduino Uno.....	15
2.6.	Analisa Data Pengukuran Vibrasi	16
2.6.1.	Ketidakeimbangan (Unbalance)	16
2.6.2.	Pompa dan Motor Yang Tidak Selurus (Pump and Motor Misalignment)	17
2.6.3.	Identifikasi Model-model Vibrasi Kondisi <i>Unfault</i> dan <i>Fault</i>	18
2.7.	Fast Fourier Transform.....	19
2.8.	Neural Network (Jaringan Saraf Tiruan).....	21
BAB 3 PERANCANGAN SISTEM		25
3.1.	Sistem Identifikasi Vibrasi Saat Melakukan Pengukuran On-line.....	25
3.1.1.	Sensor Accelerometer.....	25
3.1.2.	Raspberry Pi	26
3.1.3.	FFT (Fast Fourier Transform)	27
3.1.4.	Neural Network	30
3.2.	Sistem Pembelajaran Neural Network	33
3.2.1.	Pengambilan Data Frekuensi Untuk ke-4 Pola Vibrasi	33
3.2.2.	Pembelajaran dengan Lazarus	34
3.3.	Metode Pengambilan Data	37
3.3.1.	Model Kondisi Unfault (Kondisi normal)	37
3.3.2.	Model Kondisi Unbalance (Ketidakseimbangan).....	37

3.3.3.	Model Kondisi Misalignment (Tidak Selurus)	39
3.3.4.	Model Kondisi Looseness (Kondisi base kendur)	40
BAB 4 HASIL DAN ANALISA DATA		41
4.1.	Pola-pola Spektrum Vibrasi pada Prototipe Motor Induksi ..	41
4.1.1.	Spektrum Kondisi <i>Unfault</i> (Normal)	43
4.1.2.	Spektrum Kondisi <i>Unbalance</i> (Ketidakseimbangan).....	43
4.1.3.	Spektrum Kondisi <i>Misalignment</i> (Tidak selurus)	45
4.1.4.	Spektrum Kondisi <i>Looseness</i> (Kendur/longgar)	46
4.2.	Klasifikasi Menggunakan Neural Network	47
BAB 5 PENUTUP		53
5.1.	Kesimpulan.....	53
5.2.	Saran.....	53
DAFTAR PUSATAKA		55
LAMPIRAN		57
RIWAYAT HIDUP		77



Halaman ini sengaja dikosongkan

DAFTAR TABEL

Tabel 3.1 Tabel hubungan pola vibrasi dengan NN 31

Tabel 4.1 Hasil Pengujian NN 52



Halaman ini sengaja dikosongkan

DAFTAR GAMBAR

Gambar 1.1 Motor Induksi	1
Gambar 2.1 Motor Induksi	6
Gambar 2.2 Gerak bebas Spring yang berosilasi	8
Gambar 2.3 Struktur MEMS	11
Gambar 2.4 Sensor ADXL345 dan blok diagramnya	13
Gambar 2.5 Raspberry Pi model B+	14
Gambar 2.6 Spesifikasi Raspberry Pi B+	14
Gambar 2.7 Arduino Uno Modul	15
Gambar 2.8 Lokasi pengukuran vibrasi dan pembacaan FFT yang menunjukkan puncak vibrasi karena unbalance.....	17
Gambar 2.9 Misalignment angular	17
Gambar 2.10 Misalignment Paralel.....	18
Gambar 2.11 Spektrum vibrasi untuk kondisi unfault dan beberapa kondisi fault	19
Gambar 2.12 Spektrum frekuensi kondisi kerusakan bearing pada outer ball pass (OBPF).....	19
Gambar 2.13 Sinyal fungsi waktu hasil pengukuran vibrasi	20
Gambar 2.14 Sinyal sinus dalam domain waktu dan frekuensi.....	21
Gambar 2.15 Jaringan saraf dengan 3 lapisan	22
Gambar 2.16 Arsitektur Backpropagation	23
Gambar 3.1 Blok diagram sistem vibrasi pengukuran on-line	25
Gambar 3.2 GPIO Raspberry dan Arduino	26
Gambar 3.3 Sinyal domain waktu keluaran accelerometer	27
Gambar 3.4 Flow chart proses FFT dengan sample data vibrasi 1000 data	29

Gambar 3.5 Grafik sinyal dalam domain frekuensi	30
Gambar 3.6 Grafik domain frekuensi	30
Gambar 3.7 Arsitektur neural network dalam penelitian ini	31
Gambar 3.8 Flow chart proses neural network untuk identifikasi pola vibrasi	33
Gambar 3.9 Blok diagram proses pembelajaran NN dengan Lazarus	34
Gambar 3.10 Flow chart proses pembelajaran NN dengan Lazarus ...	35
Gambar 3.11 Hubungan Kopling pompa dan motor saat Unfault.....	37
Gambar 3.12 Posisi baut pada prototype base pompa dan motor	38
Gambar 3.13 Base motor dan pompa saat Unbalance kondisi 1	38
Gambar 3.14 Base motor dan pompa saat Unbalance kondisi 2	39
Gambar 3.15 Base motor dan pompa saat Unbalance kondisi 3	39
Gambar 3.16 Base motor dan pompa saat Unbalance kondisi 4	39
Gambar 3.17 Model kondisi Misalignment (Tidak Selurus)	40
Gambar 4.1 Foto rangkaian prototipe saat pengambilan data	41
Gambar 4.2 Tampilan sinyal pembuktian FFT	42
Gambar 4.3 Domain waktu kondisi unfault	42
Gambar 4.4 Spektrum frekuensi kondisi normal	43
Gambar 4.5 Spektrum frekuensi kondisi normal 2	43
Gambar 4.6 Spektrum frekuensi unbalance kondisi 1	44
Gambar 4.7 Spektrum frekuensi unbalance kondisi 2	44
Gambar 4.8 Spektrum frekuensi unbalance kondisi 3	44
Gambar 4.9 Spektrum frekuensi unbalance kondisi 4	44
Gambar 4.10 Spektrum frekuensi misalignment 1	46
Gambar 4.11 Spektrum frekuensi misalignment 2	46

Gambar 4.12 Spektrum frekuensi looseness kondisi 1	47
Gambar 4.13 Spektrum frekuensi looseness kondisi 2	47
Gambar 4.14 Proses pembelajaran NN di laptop	47
Gambar 4.15 Hasil neural network kondisi unfault	48
Gambar 4.16 Spektrum vibrasi kondisi unfault	48
Gambar 4.17 Hasil neural network kondisi unbalance	49
Gambar 4.18 Spektrum vibrasi kondisi unbalance	49
Gambar 4.19 Hasil neural network kondisi misalignment	50
Gambar 4.20 Spektrum vibrasi kondisi misalignment	50
Gambar 4.21 Hasil neural network kondisi looseness	51
Gambar 4.22 Spektrum vibrasi kondisi looseness	51

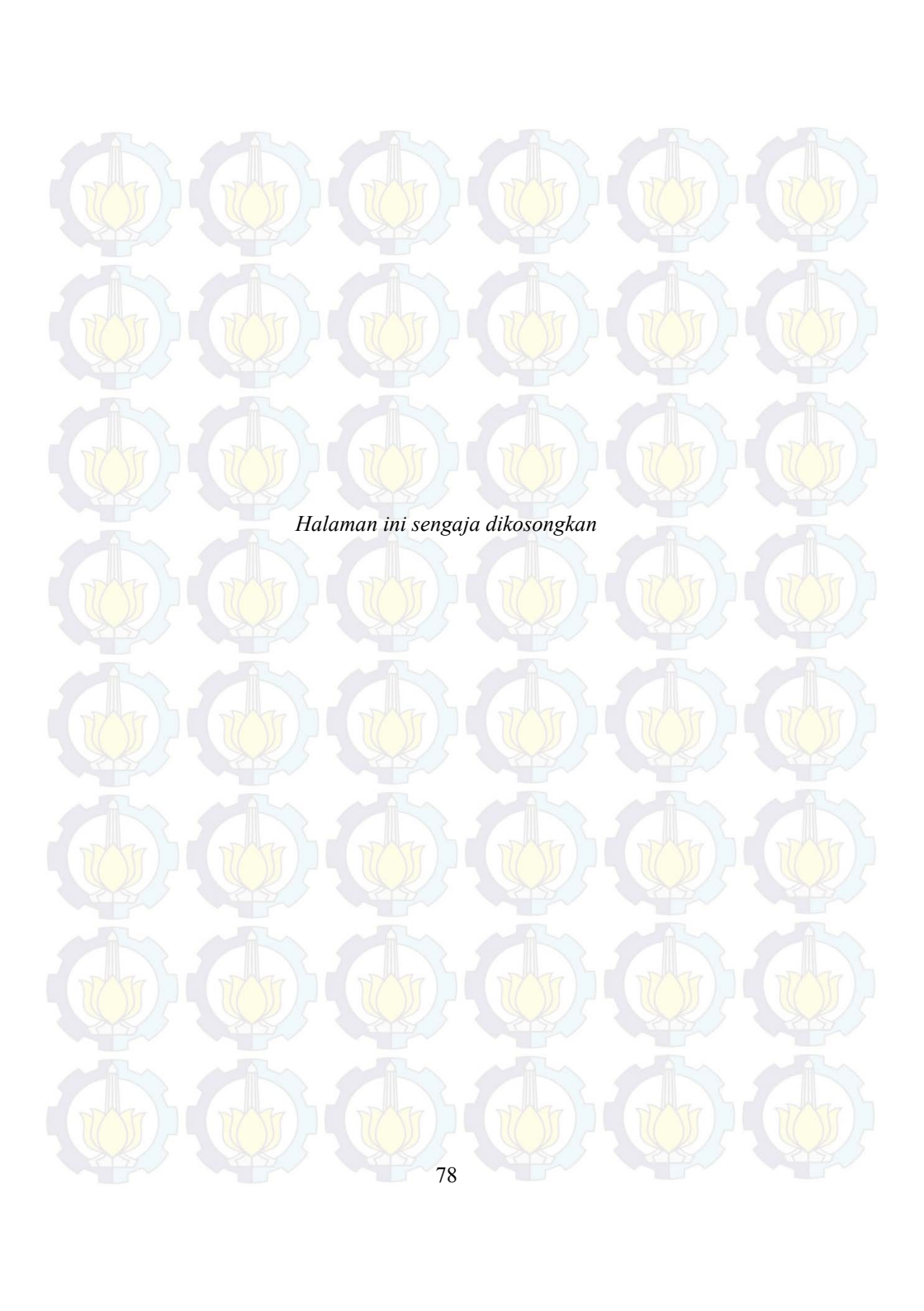


Halaman ini sengaja dikosongkan

RIWAYAT HIDUP



Rahmadi Ridho Anjiko dilahirkan di Nganjuk, 12 Juni 1992. Merupakan putra pertama dari tiga bersaudara pasangan Bapak Sunaryo dan Ibu Siti Samsiyah. Penulis menempuh jenjang pendidikan dari SD Payaman 2 Nganjuk, dan melanjutkan ke SMPN 1 Nganjuk. Kemudian penulis melanjutkan ke SMAN 2 Nganjuk hingga lulus tahun 2011. Pada tahun yang sama penulis melanjutkan studinya ke Institut Teknologi Sepuluh Nopember Surabaya Fakultas Teknologi Industri Jurusan Teknik Elektro dan mengambil bidang studi Elektronika. Penulis aktif dalam kepanitian acara Electrical Engineering Event yang meliputi LCEN, SITIA, dan Baronas 2014. Penulis juga aktif dalam kegiatan seminar dan pelatihan yang diselenggarakan di lingkungan ITS. Penulis dapat dihubungi melalui email : ridhoanjiko@gmail.com



Halaman ini sengaja dikosongkan

BAB 1

PENDAHULUAN

1.1. Latar Belakang

Dalam kehidupan sehari-hari kita motor induksi banyak digunakan di bidang industri maupun di rumah tangga. Hal ini disebabkan karena motor induksi memiliki konstruksi yang sederhana. Selain karena konstruksinya yang sederhana motor induksi juga mempunyai harga yang relative murah dan mudah dalam pengoperasian dan perawatannya. Motor induksi ini juga mudah didapatkan di toko-toko hardware.

Motor induksi yang umum dipakai adalah motor induksi tiga fasa dan motor induksi satu fasa. Motor induksi tiga fasa dioperasikan pada sistem tiga fasa dan banyak digunakan di industri, sedangkan motor induksi satu fasa dioperasikan pada sistem satu fasa yang banyak digunakan terutama pada penggunaan untuk peralatan rumah tangga seperti pompa air, mesin cuci dan sebagainya. Bentuk fisik dari motor induksi dapat dilihat pada gambar berikut ini



Gambar 1.1 Motor Induksi

Berbagai cara dilakukan untuk memelihara atau merawat motor agar motor selalu dalam keadaan prima. Selain itu juga untuk memperpanjang umur pemakaian motor induksi. Tindakan preventif harus dilakukan untuk mencegah kerusakan yang lebih besar pada

mesin. Salah satu tindakan preventif yang dapat dilakukan adalah monitoring vibrasi pada motor.

Dalam penerapannya di industri base motor akan mengalami pergeseran, namun hal itu tidak akan terjadi secara mendadak, hal ini memerlukan waktu. Sistem vibrasi yang terpasang tidak dapat mengetahui lebih awal jenis atau model kondisi abnormal yang akan terjadi pada motor. Dengan mengetahui lebih awal jenis kondisi abnormal pada pompa maka akan membantu dalam menentukan jenis tindakan pencegahan untuk melindungi pompa dan motor dari kerusakan yang lebih parah.

Dari sisi produksi bila suatu motor tidak dideteksi kondisinya secara kontinyu maka bila suatu saat motor tersebut rusak maka proses produksi perusahaan tersebut akan terganggu. Sehingga otomatis perusahaan tersebut mengalami kerugian.

Hal hal inilah yang melatarbelakangi penulis untuk melakukan penelitian untuk melakukan penelitian ini. Diharapkan dari hasil penelitian ini penulis dapat menemukan suatu sistem tertentu untuk mendeteksi vibrasi atau getaran pada motor induksi sehingga dapat melindungi motor dari kerusakan yang lebih parah. Jika sistem ini ditemukan, maka banyak hal yang dapat dilindungi pada motor diantaranya meningkatkan kehandalan motor, meningkatkan lifetime motor, dan menurunkan biaya pemeliharaan. Algoritma yang akan digunakan adalah fast fourier transform (FFT) dan neural network.

1.2. Perumusan Masalah

Berdasarkan latar belakang di atas, dapat dirumuskan beberapa masalah, antara lain :

1. Bagaimana sensor accelerometer ini mendeteksi vibrasi yang terjadi pada motor induksi.
2. Bagaiman mengetahui pola spektrum vibrasi pada kondisi fault (abnormal) dan unfault (normal).
3. Bagaiman mengimplementasikan sistem ini dalam raspberry.

1.3. Batasan Permasalahan

Untuk menyelesaikan masalah dalam Tugas Akhir ini, maka perlu diberi batasan - batasan dan asumsi sebagai berikut:

1. Membuat program FFT(Fast Fourier Transform) dan Neural Network untuk mendeteksi secara langsung dengan Lazarus di Raspberry Pi, sementara untuk training bobotnya di Lazarus laptop.
2. Penelitian ini hanya mendeteksi motor apakah dalam kondisi unfault, unbalance, misalignment, looseness.

1.4. Tujuan Permasalahan

Tujuan yang ingin dicapai dalam tugas akhir ini adalah sebagai berikut:

3. Untuk mengidentifikasi kerusakan motor melalui spektrum vibrasi yang ditimbulkan.
4. Untuk memonitoring keadaan motor listrik secara kontinyu sehingga keadaan abnormal pada motor listrik dapat diketahui secara dini.

1.5. Metode Penelitian

Metedologi yang digunakan dalam tugas akhir ini adalah sebagai berikut:

5. Studi Literatur
Literatur yang digunakan berasal dari jurnal, prosiding, text book, dan tugas akhir sebelumnya. Dalam studi literatur dipelajari teori pendukung yang berisi tentang pengeretian vibrasi dan bagaimana karakteristik vibrasi, motor listrik dan cara kerjanya, jenis jenis sensor rotating machinery serta cara kerjanya.
6. Perancangan
Penulis menyiapkan motor induksi yang akan dibuat objek penelitian, mencari rumus FFT dan NN yang akan dipakai dalam program yang akan dibuat.
7. Pembuatan Program
Pembuatan program untuk melakukan pendeteksian jenis jenis vibrasi dari motor induksi berdasarkan literatur yang telah dipelajari dan diterapkan pada Lazarus di Raspberry Pi.

8. Analisis Data

Simulasi menggunakan fungsi objektif yang telah ditentukan dan melakukan analisa terhadap jenis jenis kondisi dari motor induksi tersebut.

9. Penulisan Buku

Hasil penelitian yang telah dilakukan dilaporkan dalam bentuk laporan Tugas Akhir dan dikembangkan dalam bentuk paper. Laporan ditulis berdasarkan kesimpulan dari permasalahan yang dianalisis beserta tahapan yang ada di dalamnya.

1.6. Sistematika Penulisan

Untuk mempermudah pembahasan, dalam penulisan tugas akhir ini akan dibagi menjadi lima bab dan beberapa subbab dengan sistematika sebagai berikut:

Bab 1: Pendahuluan

Bab ini membahas mengenai latar belakang, tujuan, rumusan masalah, batasan masalah, manfaat penelitian.

Bab 2: Tinjauan Pustaka

Bab ini membahas teori penunjang mengenai motor induksi, analisa vibrasi, motor induksi, getaran, Raspberry Pi dan Arduino Uno, analisa data pengukuran vibrasi, Fast Fourier Transform (FFT), Neural Network (NN).

Bab 3: Metodologi Penelitian

Bab ini menguraikan sistem identifikasi vibrasi saat melakukan pengukuran on-line, sistem pembelajaran neural network, metode pengambilan data.

Bab 4: Hasil dan Analisa Data

Bab ini berisi tentang pola pola spektrum vibrasi pada motor, klasifikasi neural network.

Bab 5: Kesimpulan dan Saran

Bab ini merupakan bab terakhir yang berisi kesimpulan yang diperoleh dari hasil simulasi dan analisis pada tugas akhir dan saran untuk pengembangan lebih lanjut.

BAB 2

TINJAUAN PUSTAKA

2.1. Analisa Vibrasi

2.1.1. Analisa Vibrasi dan noise pada motor induksi – sumber vibrasi dan metode diagnosanya

Paper ini dibuat Ravindra Birajdar pada konferensi Internasional ke-3 tentang Integrity, Realiability, dan Failure di Portugal pada 20-24 juli 2009. Pada paper ini dijelaskan tentang sumber sumber tentang sumber sumber vibrasi dan noise pada motor dan pompa sentrifugal. Juga diberikan metode diganosa gelombang vibrasi dengan menggunakan Fast Fourier Transform untuk mendapatkan spektrum vibrasi.

Pada penelitian ini, motor yang akan dijadikan objek penelitian adlah motor induksi satu phasa. Metode diagnose yang digunakan adalah sama yaitu dengan menggunakan FFT. Yang pada akhirnya dengan selesainya penelitian ini, penulis dapat membandingkan spektrum vibrasi motor induksi.

2.1.2. Dasar-dasar monitor vibrasi untuk mendeteksi kegagalan dan control proses

Paper ini dibuat oleh Wilfried Reimche di Universitas Hannover, Institut Teknik material di Rio de Janeiro, Brasil pada 2-6 juni 2003. Paper ini menjelaskan bagaimana menganalisa dan memonitor vibrasi pada peralatan-peralatan mekanik yang komplek misalnya deteksi vibrasi pada turbin uap dan pada motor pada pompa air dengan kapasitas besar (di atas 1000 HP). Analisa yang digunakan juga FFT untuk mendapatkan spektrum vibrasi. Di paper ini tidak disebutkan secara detail jenis pompa atau turbin yang dianalisa.

Sementara pada penelitian ini, analisa vibrasi terfokus motor induksinya.

2.1.3. Analizer FFT untuk diagnose kegagalan pada mesin berputar berdasarkan analisa vibrasi

Paper ini dibuat oleh Giovanni Beta (IEEE, 2002) yang melakukan penelitian pada motor listrik tiga phase. Pada penelitian ini, dilakukan pemodelan untuk beberapa jenis kondisi *fault* dan *unfault*.

Pada penelitian ini, pengukuran dilakukan pada motor yang dihubungkan pompa sentrifugal.

2.2. Motor Induksi

Motor induksi sangat banyak digunakan dalam kehidupan sehari-hari baik di industri maupun di rumah tangga karena konstruksinya sederhana, harganya relative murah dan mudah dalam pengoperasian dan perawatan. Motor induksi yang umum dipakai adalah motor induksi tiga phasa dan motor induksi satu phasa. Motor induksi tiga phasa dioperasikan pada sistem tiga phasa dan banyak digunakan di industri, sedangkan motor induksi satu phasa dioperasikan pada sistem satu phasa yang banyak digunakan terutama pada penggunaan untuk peralatan rumah tangga seperti pompa air, mesin cuci dan sebagainya. Bentuk fisik dari motor induksi dapat dilihat pada gambar berikut ini:



Gambar 2.1 Motor Induksi

Motor ini bekerja berdasarkan induksi medan magnet stator ke rotornya. Ketika Tegangan AC 3 phasa yang diberikan ke stator menghasilkan medan putar (*rotating magnetic field*) pada stator. Medan putar tersebut memotong konduktor pada rotor sehingga menghasilkan

induksi ggl (gaya gerak listrik) pada rotor (berdasarkan hukum Faraday). Karena kumparan rotor merupakan rangkaian tertutup maka ggl akan menghasilkan arus I. Adanya arus di dalam medan magnet menimbulkan gaya F pada rotor sehingga rotor berputar.

$$ns = \frac{120f}{p}$$

Dimana :

ns = kecepatan sinkron [rpm]

f = frekwensi sumber [Hz]

p = jumlah kutub

Jika motor induksi bekerja pada frekuensi 50 Hz dengan jumlah kutub 2 maka kecepatan putar motor tersebut adalah $ns = 120.50/2 = 3000$ rpm. Perbedaan antara kecepatan sinkron dan kecepatan rotor disebut slip.

$$s = \frac{ns - n}{ns}$$

Dimana :

s = slip

ns = kecepatan sinkron [r/min]

n = kecepatan rotor [r/min]

2.3. Getaran

Getaran adalah merupakan gerakan osilasi pada sistem mekanik pada titik referensi. Getaran sendiri dapat dibagi menjadi dua bagian besar yaitu getaran yang periodic maupun acak.

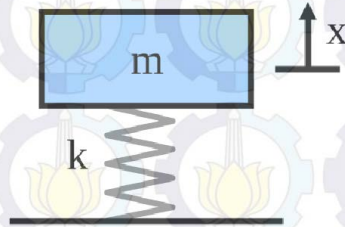
Pada getaran yang berperiodik, maka gerakan akan berulang-ulang pada interval waktu tertentu, demikian yang non periodic tidak berulang-ulang pada interval waktu tertentu. Getaran sendiri dapat dibagi menjadi dua bagian besar yaitu getaran yang periodic maupun acak.

Pada getaran yang berperiodik, maka gerakan akan berulang-ulang pada interval waktu tertentu, demikian yang non periodic tidak berulang-ulang pada interval waktu tertentu.

Getaran pada umumnya ada yang diinginkan ada yang tidak, pada getaran yang diinginkan seperti garputala maupun speaker. Namun

ada getaran yang tidak diinginkan, karena dapat melemahkan daya, akan menimbulkan gangguan suara (Noise), dan mengidentifikasi adanya ketidak normalan dari sistem mekanik yang bergetar.

Prinsip dasar getaran dapat digambarkan sebagai sebuah gerak bebas dari per (Spring). Seperti gambar dibawah ini.



Gambar 2.2 Gerak bebas Spring yang berosilasi

Dimana persamaan untuk gerak bebas sebuah per (spring) adalah: $F_s = -k \cdot x$ dimana persamaan x dapat dituliskan sebagai

$$X(t) = A \sin (2\pi f_n t) = A \sin (\omega t) \quad (2.1)$$

$$f_n = \frac{1}{2\pi} \sqrt{\frac{k}{m}} \quad (2.2)$$

Persamaan 2.1 ini merupakan persamaan pergerakan dari getaran secara umum, maka turunan pertama dari pergerakan getaran adalah merupakan kecepatan dari getaran. Dan turunan kedua dari pergerakan adalah percepatan getaran.

$$v = \frac{dx}{dt} = \omega A \cos(\omega t) \quad (2.3)$$

$$a = \frac{d^2x}{dt^2} = -\omega^2 A \sin(\omega t) \quad (2.4)$$

2.3.1. Penyebab Vibrasi

Identifikasi sumber dan deteksi kerusakan pada sinyal getaran diasosiasikan dengan jenis komponen yang berhubungan dengan getaran rotasi seperti gear, rotor, dan shaft, rolling element bearing, flexible coupling dan mesin listrik.

Penyebab utama dari getaran mekanik adalah ketidakseimbangan (unbalance), ketidaklurusan (misalignment), longgar/kendur (looseness) dan sebagainya.

Vibrasi mekanik sering diukur dengan menggunakan akselerometer, tetapi juga sering menggunakan probe displacement dan sensor velocity. Disamping itu, dapat juga menggunakan analyzer vibrasi.

Hal yang penting adalah untuk mengetahui lokasi sumber vibrasi. Seperti yang dijelaskan di atas bahwa gaya dapat menyebabkan vibrasi. Jika kita mengetahui tipe-tipe gaya yang menyebabkan vibrasi maka kita dapat mengetahui bagaimana gaya-gaya ini akan menyebabkan vibrasi. Pada mesin-mesin berputar, titik-titik vibrasi selalu secara langsung terletak pada bearing mesin. Penyebabnya adalah bahwa berbagai gaya dinamik dari mesin yang berputar harus diambil dalam arah horizontal, vertical dan axial pada masing masing bearing.

Vibrasi pada motor mempunyai pengaruh besar dalam menentukan unjuk kerja dari motor itu sendiri. Pada umumnya, dengan penambahan level vibrasi maka mengidentifikasi adanya kerusakan awal pada motor. Kerusakan awal ini nantinya akan menyebabkan kerusakan fatal pada motor itu sendiri. Vibrasi juga dapat menjadi keluaran dari peralatan yang respon gaya eksitasi, seperti residual rotor unbalance, aliran fluida turbulen, tekanan, kavitasi dan atau wear pompa.

Sumber-sumber vibrasi pada motor dapat dikategorikan dalam tiga tipe yaitu: penyebab mekanik, penyebab hidrolis dan penyebab peripheral.

Penyebab mekanik meliputi:

- Komponen putaran yang tidak seimbang (Unbalanced rotating equipment)
- Impeller yang rusak dan sleeve shaft yang tidak kpsentrik
- Shaft yang bengkok
- Pompa dan motor tidak selurus

- Komponen-komponen pompa dan motor yang berekspansi karena pengaruh kenaikan suhu misalnya shaft
- Bearing yang rusak
- Pipe strain
- Desain pondasi yang tidak tepat
- Penyebab hidrolis meliputi:
 - Pompa beroperasi diluar area titik efisiensi yang terbaik
 - Resirkulasi internal
 - Adanya penguapan
 - Water hammer
 - Adanya udara terjebak dalam sistem
 - Turbelansi dalam sistem
- Penyebab peripheral meliputi:
 - Vibrasi harmonic dari equipment lain yang terdekat
 - Pompa beroperasi pada kecepatan kritis

2.3.2. Accelerometer MEMS (Micro-Electro Mechanical System)

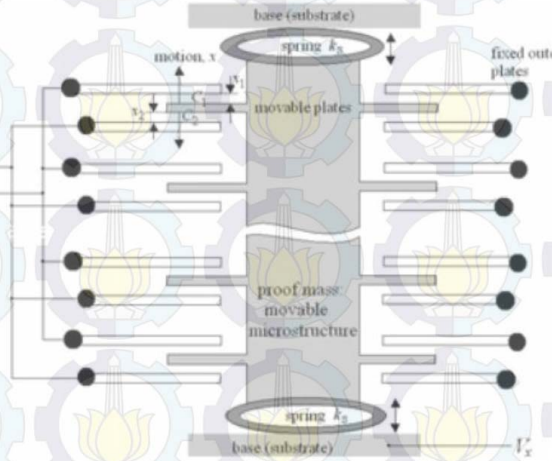
MEMS (Micro-electro Mechanical System) adalah merupakan sebuah sensor mekanik yang dikemas ke dalam bentuk Integrated Circuit (IC). Karena merupakan sebuah microelectronic maka komponen utama penyusunannya adalah silikon dan dalam ukuran micron. Dan bisa digunakan untuk mengukur percepatan, posisi, atau kejutan. Karena material yang digunakan adalah jenis Silikon, maka MEMS dapat dibuat dalam ukuran hingga micron.

Pada sensor percepatan (Accelerometer) konvensional yang ada menggunakan prinsip piezoelectric yang bentuk fisiknya lebih besar, aplikasinya meletakkan sensor ke obyek yang diukur getarannya sedikit lebih susah dan memerlukan daya yang cukup besar. Dengan beberapa kelebihan tersebut diharapkan MEMS dapat menggantikan sensor getaran yang lebih baik.

MEMS untuk percepatan memiliki struktur Capacitance untuk mendeteksi adanya perubahan percepatan (Acceleration). Dimana Capacitance akan berubah-ubah sesuai dengan pergerakan dari getaran sebagai perubahan percepatan.

Dimana ϵ adalah permitifitas bahan, dimana MEMS permitifitas yang digunakan adalah udara sehingga suhu cukup mempengaruhi besarnya permitifitas bahan. A adalah luas permukaan pelat dan d adalah

jarak antar pelat. Gambar spektrum MEMS dapat dilihat pada gambar dibawah ini.



Gambar 2.3 Struktur MEMS (Stephaen,et al, 2004)

Jika percepatan sama dengan nol(tidak ada percepatan), maka $X1 = X2 = X3$ dan nilai $C1 = C2$. Dan dimana nilai $X1$ dan $X2$ dihasilkan dari pergerakan dari pelat pembentuk kapasitansi pada MEMS yang akan berubah-ubah jika adanya percepatan yang timbul pada MEMS. Jika ada percepatan yang muncul maka nilai dari $C1$ dan $C2$ akan menjadi:

$$C1 = \epsilon \frac{1}{x1 + X} \text{ dan } C2 = \epsilon \frac{1}{x1 - x} \quad (2.5)$$

Sehingga perbedaan kapasitansinya menjadi:

$$\Delta C = C1 - C2 = 2 \epsilon \frac{x}{x^2 - x_1^2} \quad (2.6)$$

Sehingga akan didapat persamaan aljabar menjadi:

$$\Delta C x^2 - 2 \epsilon x - \Delta C x_1^2 = 0 \quad (2.7)$$

Untuk pergerakan percepatan yang terlalu kecil, maka Cx^2 dapat diabaikan. Sehingga persamaan diatas menjadi:

$$x \approx -\frac{x_1^2}{2\epsilon}\Delta C \quad (2.8)$$

Sehingga dapat dikatakan bahwa nilai pergerakan (x) akan mendekati nilai perbedaan kapasitansi C. dan menurut hukum pegas bahwa gaya F_s proporsional dengan pergerakan x dan sesuai persamaan pegas menjadi:

$$F_s = K.X \quad (2.9)$$

Dimana X merupakan konstanta pegas. Berdasarkan pada hukum Newton kedua maka dapat dituliskan:

$$m.a = m \frac{dx_1^2}{dt^2} = k.x \quad (2.10)$$

sehingga:

$$a = -\frac{k}{m}X \quad (2.11)$$

dari persamaan ini maka akan didapat:

$$a = -\frac{k.x_1^2}{2m\epsilon}\Delta C \quad (2.12)$$

dari sini dapat dikatakan bahwa percepatan (Acceleration) dapat dipengaruhi oleh besarnya perbedaan kapasitansi dari dalam MEMS.

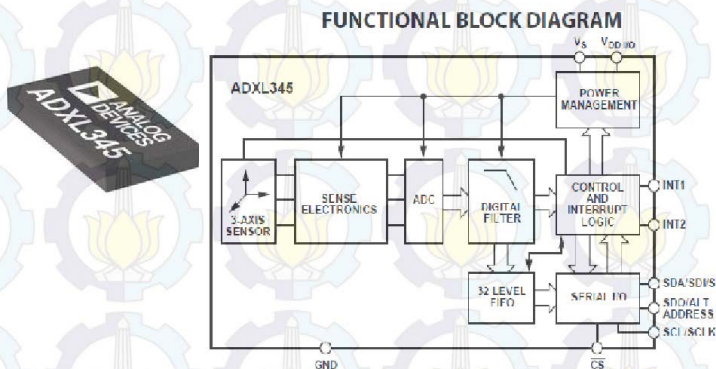
2.3.3. Accelerometer ADXL345 MEMS

Di dunia industry, sensor vibrasi pada umumnya yang digunakan ada dua tipe yaitu tipe proximity dan tipe akselerometer. Tipe proximity digunakan untuk mengukur pergeseran benda-benda yang berputar misalnya pergeseran shaft. Tipe proximity ini tidak bersentuhan langsung dengan obyek yang diukur. Sementara tipe akselerometer ini digunakan untuk mengukur vibrasi benda-benda yang statistic seperti mengukur vibrasi pada rumah bearing. Tipe akselerometer ini bersentuhan langsung dengan obyek yang diukur.

Sensor yang digunakan dalam penelitian ini adalah sensor akselerometer ADXL345. ADXL345 ini memiliki jangkauan pengukuran minimal $\pm 16g$ (gravitasi). Sinyal keluaran tegangan analog yang

dihasilkan oleh sensor ini berbanding lurus dengan percepatan. Akselerometer dapat mengukur percepatan statis gravitasi dalam aplikasi deteksi kemiringan dan juga mengukur percepatan dinamis yang dihasilkan dari gerakan, kejutan/gonjangan atau getaran.

Akselerometer ADXL345 adalah sejenis Micro-Electro-Mechanical Sensor (MEMS). Dari namanya ada hubungan dengan benda mekanik dan sensor elektronik didalamnya. Disebut mikro karena ukuran komponen dalam chips seukuran 0,001 – 0,1 mm. Prinsip kerja yang digunakan adalah prinsip percepatan (acceleration). Sebuah per dengan beban dan diepaskan, beban bergerak dengan suatu percepatan sampai kondisi tertentu akan berhenti. Bila ada sesuatu yang menggoncangkannya maka beban akan berayun kembali. Konsep ini diterapkan pada sebuah chip kecil, dimana per digantikan dengan plat mini yang ujungnya diberi penyangga dan ditengahnya diberi beban. Lain pelat dipasang dalam posisi tetap berhadapan. Saat ada gaya dari luar maka pelat beban akan bergerak mendekat pelat tetap, bila pelat sebagai kapasitor maka besaran kapasitansi chip MEMS. Agar sensor ini bisa mendeteksi 3 dimensi, maka dibutuhkan 3 pasang plat yang dipasang tegak lurus antar masing masing.



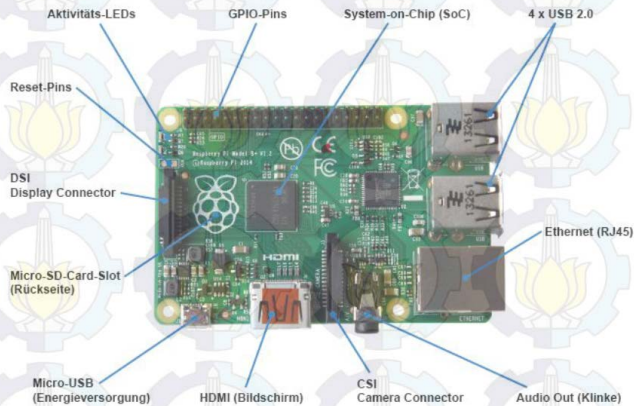
Gambar 2.4 Sensor ADXL345 dan blok diagramnya

2.4. Raspberry Pi

Raspberry Pi (juga dikenal sebagai RasPi) adalah sebuah SBC (Single-Board Computer) seukuran kartu kredit yang dikembangkan

oleh Yayasan Raspberry Pi di Inggris (UK) dengan maksud untuk memicu pengajaran ilmu komputer dasar di sekolah-sekolah.

Raspberry Pi menggunakan system on a chip (SoC) dari Broadcom BCM2835, juga sudah termasuk prosesor ARM1176JZF-S 700 MHz, GPU VideoCore IV dan RAM sebesar 256 MB (untuk Rev. B). Tidak menggunakan hard disk, namun menggunakan SD Card untuk proses booting dan penyimpanan data jangka-panjang.



Gambar 2.5 Raspberry Pi model B+

Specifications

Chip	Broadcom BCM2835 SoC
Core architecture	ARM11
CPU	700 MHz Low Power ARM1176JZF8 Applications Processor
GPU	Dual Core VideoCore IV® Multimedia Co-Processor Provides Open GL ES 2.0, hardware-accelerated OpenVG, and 1080p30 H.264 high-profile decode Capable of 1Gpixel/s, 1.5Gstixel/s or 24GFLOPs with texture filtering and DMA Infrastructure
Memory	512MB SDRAM
Operating System	Boots from Micro SD card, running a version of the Linux operating system
Dimensions	85 x 56 x 17mm
Power	Micro USB socket 5V, 2A

Gambar 2.6 Spesifikasi Raspberry Pi B+

2.5. Arduino Uno

Arduino Uno adalah papan mikrokontroler berbasis ATmega328. Arduino *Uno* memiliki 14 digital pin input/output, dimana 6 pin digunakan sebagai output PWM, 6 pin *input* analog, 16 MHz *resonator* keramik, koneksi USB, *jack* catu daya eksternal, header ICSP, dan tombol *reset*. Ini semua berisi hal-hal yang diperlukan untuk mendukung mikrokontroler, sederhana saja, hanya dengan menghubungkannya ke komputer dengan kabel USB atau sumber tegangan dengan adaptor AC-DC dan atau baterai untuk memulai menggunakan papan Arduino.

Arduino Uno R3 berbeda dari semua papan Uno sebelumnya yang sudah tidak menggunakan *chip driver* FTDI USB-to-serial. Sekarang, Arduino Uno menggunakan fitur Atmega16U2 (Atmega8U2 sampai dengan versi R2) yang diprogram sebagai konverter USB-to-serial.



Gambar 2.7 Arduino Uno Modul

Arduino Uno dapat diaktifkan melalui koneksi USB atau dengan catu daya eksternal. Sumber daya akan dipilih secara otomatis oleh Arduino. Sumber daya eksternal (non-USB) dapat berasal baik dari adaptor AC-DC atau baterai. Adaptor dapat dihubungkan dengan mencolokkan steker 2,1 mm yang bagian tengahnya terminal positif ke ke *jack* sumber tegangan pada papan. Jika tegangan berasal dari baterai dapat langsung dihubungkan melalui *header* pin Gnd dan pin Vin dari konektor POWER. Papan Arduino Uno dapat beroperasi dengan daya eksternal 6 Volt sampai 20 volt. Jika diberi tegangan kurang dari 7 Volt, maka, pin 5 Volt mungkin akan menghasilkan tegangan kurang dari 5 Volt dan ini akan membuat papan menjadi tidak stabil. Jika sumber

tegangan menggunakan lebih dari 12 Volt, *regulator* tegangan akan mengalami panas berlebihan dan bisa merusak papan. Rentang sumber tegangan yang dianjurkan adalah 7 Volt sampai 12 Volt. Arduino disini saya gunakan untuk membaca sensor ADXL345 yang menggunakan SPI communication, kemudian hasil baca sensornya baru saya olah di raspberry. Langkah ini digunakan karena lazarus di raspberry tidak bisa membaca pin input SPI langsung dari sensor ADXL345, sehingga ditempuh cara ini dengan membuat data menjadi biner terlebih dahulu, karena setelah di konversikan baru bisa dibaca oleh lazarus melalui pin GPIO yang tersedia di raspberry.

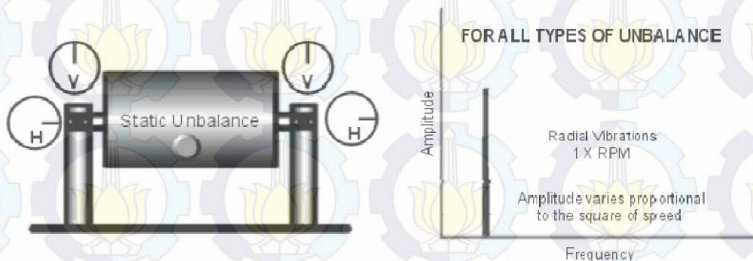
2.6. Analisa Data Pengukuran Vibrasi

Ada berbagai metode yang digunakan dalam menganalisa data-data vibrasi. Metode dasar dengan menampilkan data vibrasi dalam area frekuensi, juga disebut dengan spektrum vibrasi. Frekuensi vibrasi adalah jumlah siklus vibrasi per satuan waktu. Spektrum vibrasi adalah bagian dasar dalam menganalisa vibrasi, karena dapat menampilkan informasi-informasi yang tersembunyi pada bentuk gelombang vibrasi berdasarkan waktu. Untuk mendapatkan spektrum vibrasi dapat dilakukan dengan berbagai cara, yang sangat sering digunakan adalah Fast Fourier Transform (FFT).

Dari grafik dalam bentuk gelombang FFT, dapat didiagnosa penyebab vibrasi dan dapat juga dilihat model model vibrasi. Berikut ini adalah beberapa model vibrasi yang umum terjadi.

2.6.1. Ketidakeimbangan (Unbalance)

Ketidakeimbangan untuk seterusnya akan kita sebut *unbalance*, FFT spektrum akan menunjukan puncak gelombang pada frekuensi 1x rpm vibrasi dan amplitude pada frekuensi 1x rpm akan bervariasi proposional terhadap kuadrat kecepatan putaran. Jika *unbalance* terjadi maka puncak gelombang pada frekuensi 1x rpm mendominasi spektrum vibrasi.

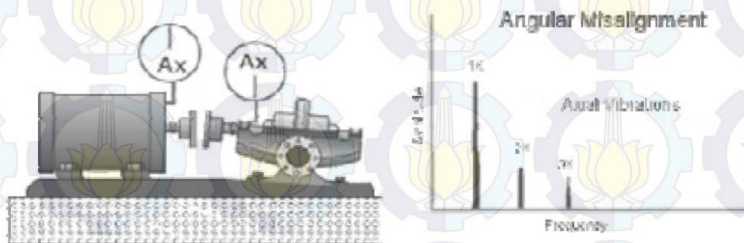


Gambar 2.8 Lokasi pengukuran vibrasi dan pembacaan FFT yang menunjukkan puncak vibrasi karena unbalance (Birajdar, Ravindra, 2009)

2.6.2. Pompa dan Motor Yang Tidak Selurus (Pump and Motor Misalignment)

Istilah tidak selurus selanjutnya akan diganti dengan *misalignment*. Pada dasarnya ada 2 tipe *misalignment* yang terjadi diantara pompa dan motor:

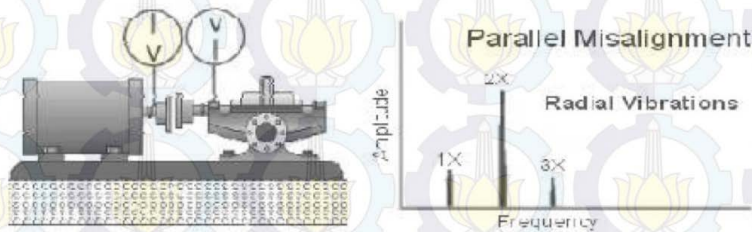
- *Misalignment* angular – yaitu garis tengah kedua shaft (shaft motor dan shaft pompa) membentuk suatu sudut
- *Misalignment* parallel – yaitu garis tengah kedua shaft (shaft motor dan shaft pompa) parallel.



Gambar 2.9 Misalignment angular (Birajdar, Ravindra, 2009)

Seperti yang ditunjukkan pada gambar 2.8 bahwa misalignment angular akan menyebabkan puncak gelombang vibrasi axial pada frekuensi 1x rpm. Secara umum pada misalignment jenis ini, vibrasi axial dapat juga dilihat pada frekuensi 1x rpm, 2x rpm, 3 x rpm.

Untuk misalignment parallel dapat dilihat pada gambar dibawah ini.



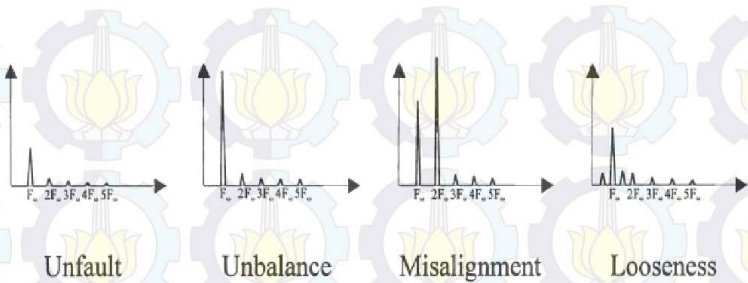
Gambar 2.10 Misalignment Paralel (Birajdar, Ravindra, 2009)

Misalignment parallel akan menghasilkan spektrum vibrasi dengan puncak vibrasi radial pada frekuensi 2x rpm. Vibrasi radial pada frekuensi 1x rpm dan 3x rpm tetap muncul, tetapi lebih kecil dari frekuensi 2x rpm.

2.6.3. Identifikasi Model-model Vibrasi Kondisi *Unfault* dan *Fault*

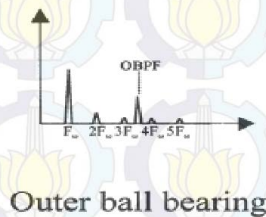
Pendekatan model berbasis diagnostik memerlukan identifikasi dari model *unfault* (normal) dan *faults* (tidak normal). Dalam analisis vibrasi frekuensi-domain, model ini dapat terdiri dari parameter yang sesuai dari spektrum getaran sinyal. Spektrum getaran selama kondisi normal ditandai dengan amplitude tertinggi terletak pada frekuensi putaran motor, diikuti oleh sejumlah frekuensi harmonik yang amplitudanya kurang dari satu-sepertiga amplitude frekuensi putaran (lihat gambar 2.9). Setiap jenis kegagalan mekanis menyebabkan perubahan tertentu dari spektrum vibrasi yang kemudian disebut kondisi tidak normal. Berikut ini, beberapa contoh model vibrasi tidak normal yang paling umum terjadi.

sebuah kondisi *unbalance* (ketidakseimbangan poros) menyebabkan amplitude vibrasi radial tertinggi pada frekuensi putaran. Untuk kondisi *misalignment* amplitude tertinggi akan terjadi pada frekuensi-frekuensi harmonik (dua sampai lima kali frekuensi putaran). Semakin tinggi nilai amplitude pada frekuensi harmonik menunjukkan semakin tinggi tingkat kerusakan kondisi *misalignment*. Sementara ini, untuk kondisi (longgar/kendur), hal ini akan menyebabkan bertambahnya frekuensi harmonik dan akan menyebabkan juga semakin tinggi nilai amplitude pada frekuensi putaran jika dibandingkan dengan kondisi normal.



Gambar 2.11 Spektrum vibrasi untuk kondisi unfault dan beberapa kondisi fault (Giovanni Betta, IEEE, 2002)

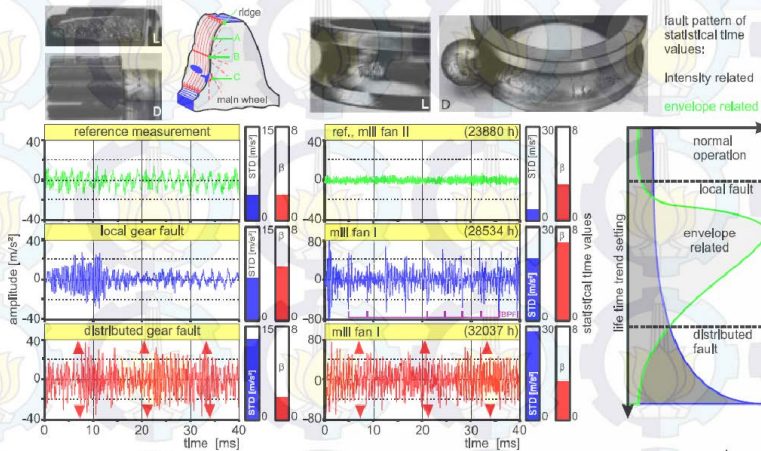
Selain kondisi diatas, kerusakan *bearing* juga akan membangkitkan suatu frekuensi harmonik diluar frekuensi harmonik putaran poros motor. Frekuensi-frekuensi yang akan muncul diantaranya frekuensi kecepatan putaran, *outer* dan *inner ball pass frequencies* (IBPF dab OBPF) dan frekuensi putaran ball. Berikut ini adalah spektrum frekuensi untuk kondisi OBPF.



Gambar 2.12 Spektrum frekuensi kondisi kerusakan bearing pada outer ball pass (OBPF)

2.7. Fast Fourier Transform

Sinyal vibrasi yang diukur oleh transducer masih sebagai fungsi waktu dengan satuan amplitudo tertentu. Berikut ini adalah contoh hasil pengukuran fungsi waktu vibrasi.



Gambar 2.13 Sinyal fungsi waktu hasil pengukuran vibrasi

Untuk menganalisa sinyal vibrasi diperlukan dua informasi yaitu amplitudo dan frekuensi. Untuk mendapatkan informasi ini digunakan analisa yang disebut Fast Fourier Transform. FFT ini berfungsi untuk mengubah sinyal dari *time base* menjadi *frequency base*.

Persamaan dasar dari FFT ini adalah :

$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^{kn} \quad (2.13)$$

Dengan:

n, k, N adalah integer

$K = 0, 1, 2, 3, \dots, N-1$

$$j = \sqrt{-1}$$

$$W_N = e^{-j2\pi/N} = \cos \frac{2\pi}{N} - j \sin \frac{2\pi}{N} \quad (2.14)$$

Karena $x(n) = x_r(n) + jx_i(n)$ bisa bernilai kompleks, maka

$$x(k) = x_R(k) + j X_I(k) \text{ yaitu:} \quad (2.15)$$

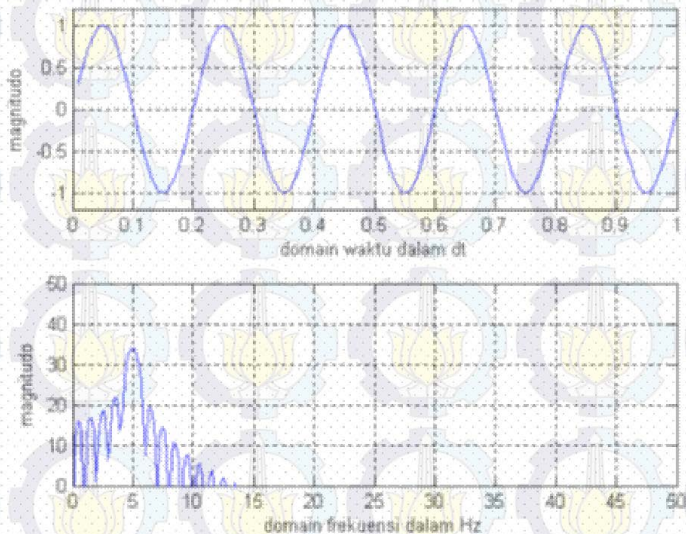
$$X_R(k) = \sum_{n=0}^{N-1} \left[Xr(n) \cos 2\pi \frac{k}{N} n + X_I(n) \sin 2\pi \frac{k}{N} n \right]$$

$$X_I(k) = \sum_{n=0}^{N-1} \left[Xr(n) \sin 2\pi \frac{k}{N} n + X_I(n) \cos 2\pi \frac{k}{N} n \right] \quad (2.16)$$

Dari persamaan ini dapat dilihat memerlukan $2N^2$ evaluasi fungsi trigonometric ditambah $4N^2$ perkalian real ditambah $4N(N-1)$ penjumlahan real dan sejumlah *indexing* dan *addressing operators*. Hal ini sering disebut dengan $O(N^2)$.

Persamaan ini merupakan bentuk transformasi fourier yang siap dikomputasi secara langsung dari bentuk sinyal $x(t)$.

Sebagai contoh, jika memiliki sinyal sinus dengan frekuensi 5 Hz dan amplitude 1 volt. Dalam domain waktu akan dapat dilihat seperti pada gambar 1 bagian atas. Sementara dalam domain frekuensi akan didapatkan seperti pada bagaian bawah.



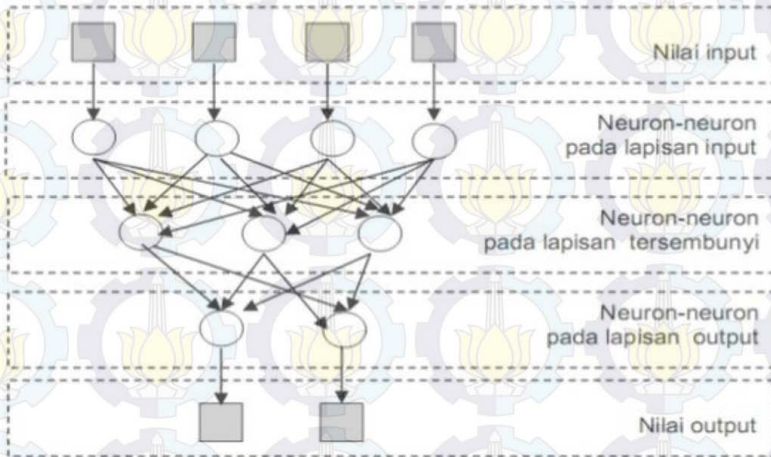
Gambar 2.14 Sinyal sinus dalam domain waktu dan frekuensi

2.8. Neural Network (Jaringan Saraf Tiruan)

Jaringan saraf tiruan (JST) atau neural network adalah suatu metode komputasinya yang meniru sistem jaringan saraf biologis. Metode ini menggunakan elemen perhitungan non-linier dasar yang disebut neuron yang diorganisasikan sebagai jaringan yang saling berhubungan, sehingga mirip dengan jaringan saraf manusia. Jaringan

saraf tiruan dibentuk untuk memecahkan suatu masalah tertentu seperti pengenalan pola atau klasifikasi karena proses pembelajaran.

Setiap pola-pola informasi input dan output yang diberikan kedalam JST diproses dalam neuron. Neuron-neuron tersebut terkumpul di dalam lapisan-lapisan yang disebut neuron layers. Lapisan-lapisan penyusun JST tersebut dapat dibagi menjadi 3, yaitu:



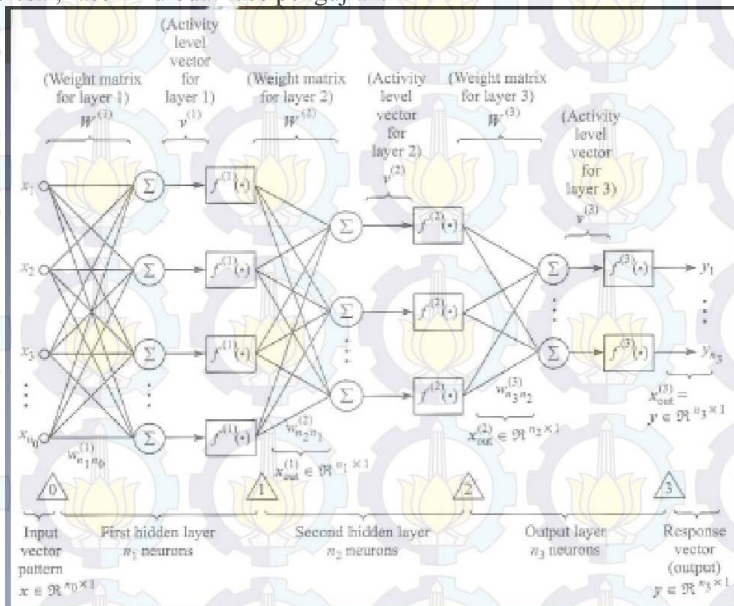
Gambar 2.15 Jaringan saraf dengan 3 lapisan

1. Lapisan Input
Unit-unit di dalam lapisan input disebut unit-unit input. Unit unit input tersebut menerima pola inputan data dari luar yang menggambarkan suatu permasalahan.
2. Lapisan tersembunyi
Unit-unit di dalam lapisan tersembunyi disebut unit unit tersembunyi. Dimana outputnya tidak dapat secara langsung diamati.
3. Lapisan output
Unit-unit di dalam lapisan tersembunyi disebut unit-unit output. Output dari lapisan ini merupakan solusi JST terhadap suatu permasalahan.

Metode algoritma yang digunakan dalam penelitian ini adalah algoritma backpropagation. Propagasi balik atau backpropagation merupakan suatu teknik pembelajaran/pelatihan supervised learning yang paling banyak digunakan. Metode ini merupakan salah satu metode

yang sangat baik dalam menangani masalah pengenalan pola-pola kompleks.

Di dalam jaringan propagasi balik, setiap unit yang berbeda di lapisan input terhubung dengan setiap unit yang ada di lapisan tersembunyi. Setiap unit yang ada di lapisan tersembunyi terhubung dengan setiap unit yang ada di lapisan output. Jaringan ini terdiri dari banyak lapisan (multilayer network). Ketika jaringan diberikan pola masukan sebagai pola pelatihan, maka pola tersebut menuju unit-unit lapisan tersembunyi untuk selanjutnya diteruskan pada unit-unit di lapisan keluaran. Kemudian unit-unit lapisan keluaran akan memberikan respon sebagai keluaran JST. Saat hasil keluaran tidak sesuai dengan yang diharapkan, maka keluaran akan disebarkan mundur (backward) pada lapisan tersembunyi kemudian dari lapisan tersembunyi menuju lapisan masukan. Tahap pelatihan ini merupakan langkah untuk melatih suatu JST, yaitu dengan cara melakukan perubahan bobot. Sedangkan penyelesaian masalah akan dilakukan jika proses pelatihan tersebut telah selesai, fase ini dibuat fase pengujian.



Gambar 2.16 Arsitektur Backpropagation

Aturan pelatihan jaringan propagasi balik terdiri dari 2 tahapan, feedforward dan backward propagation. Pada jaringan diberikan sekumpulan contoh pelatihan yang disebut set pelatihan. Set pelatihan ini diasosiasikan dengan sebuah vector feature yang disebut dengan vector input yang diasosiasikan dengan sebuah output yang menjadi target pelatihannya. Dengan kata lain set pelatihan terdiri dari vector input dan juga vector output target. Keluaran dari jaringan berupa sebuah vector output actual. Selanjutnya dilakukan perbandingan antara output actual yang dihasilkan dengan output target dengan cara melakukan pengurangan diantara kedua output tersebut. Hasil dari pengurangan merupakan error. Error dijadikan sebagai dasar dalam melakukan perubahan dari setiap bobot yang ada dengan mempropagasikan kembali. Setiap perubahan bobot yang terjadi dapat mengurangi error. Siklus setiap perubahan bobot dilakukan pada setiap set pelatihan hingga kondisi berhenti dicapai, yaitu bila mencapai jumlah bobot yang diinginkan atau hingga sebuah nilai ambang yang ditetapkan terlampaui. Algoritma pelatihan jaringan propagasi balik terdiri dari 3 tahapan yaitu:

1. Tahap umpan maju (feedforward)
2. Tahap umpan mundur (backpropagation)
3. Tahap pengupdaten bobot.

Algoritma standart backpropagation yang digunakan adalah sebagai berikut:

$$w_{ji}^{(s)}(k+1) = w_{ji}^{(s)}(k) + \mu^{(s)} \delta_j^{(s)} x_{out,i}^{(s-1)} \quad (2.17)$$

dengan,

$$\delta_j^{(s)} = (d_{qh} - x_{out,i}^{(s)}) g(v_j^{(s)}) ; \text{ untuk layer terluar} \quad (2.18)$$

$$\delta_j^{(s)} = \left(\sum_{h=0}^{s+1} \delta_h^{(s+1)} w_{hj}^{(s+1)} \right) g(v_j^{(s)}) \quad (2.19)$$

BAB 3

PERANCANGAN SISTEM

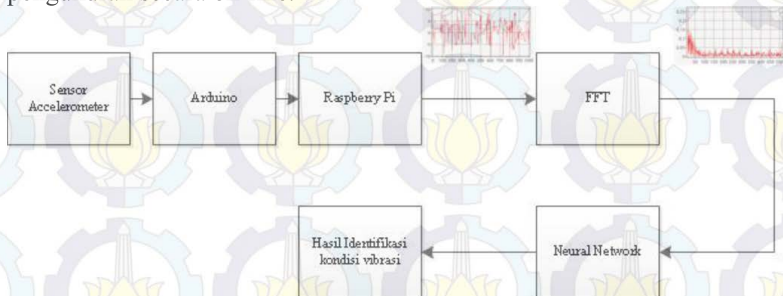
Pada penelitian ini, plant motor yang menjadi obyek penelitian merupakan plant yang berjalan kontinyu tanpa berhenti. Sehingga dibuat suatu prototype motor yang dilengkapi oleh sensor vibrasi. Dengan prototype ini dapat diteliti pola spektrum vibrasi yang terjadi motor induksi.

Metode penelitian ini dibagi dua yaitu:

1. Sistem identifikasi vibrasi saat melakukan pengukuran on-line
2. Sistem pembelajaran neural network

3.1. Sistem Identifikasi Vibrasi saat melakukan pengukuran on-line

Gambar 3.1 adalah blok diagram sistem vibrasi saat melakukan pengukuran secara on-line.



Gambar 3.1 Blok diagram sistem vibrasi pengukuran on-line

Penjelasan untuk masing masing bagian blok diagram diatas seperti dibawah ini.

3.1.1. Sensor Accelerometer

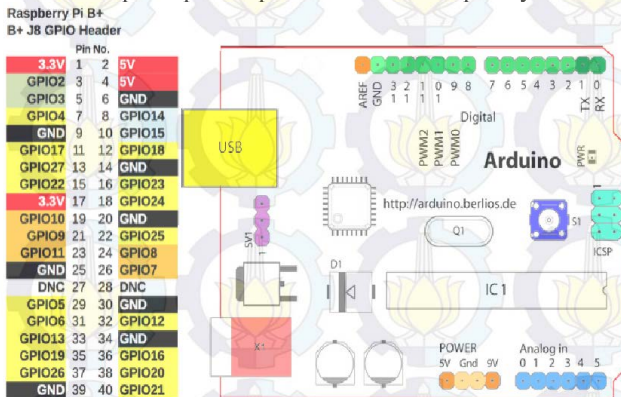
Sensor yang digunakan dalam penelitian ini adalah accelerometer ADXL345. Sensor ini mengukur percepatan dari suatu peralatan. Keluaran dari sensor berupa tegangan DC dengan jangkauan 0-3,3 VDC. Dari data sheet sensor (lampiran A), setiap 256 LSB keluaran sensor setara dengan 1g ($9,8 \text{ m/s}^2$). Sehingga:

$$1 \text{ lsb} = \frac{9,8 \text{ m/s}^2}{256} = 0.0383 \text{ m/s}^2$$

Sehingga dari perhitungan diatas dapat dilihat bahwa setiap 1 LSB setara dengan percepatan 0,0383 m/s². Sensor adxl 345 ini bisa menggunakan SPI (Serial Peripheral Interface) untuk proses komunikasinya. Di sini saya menggunakan SPI untuk komunikasi sensor dengan raspberry pi.

3.1.2. Raspberry Pi

Sistem yang digunakan untuk membaca sensor accelerometer adalah, kita terlebih dahulu menghubungkan sensor tersebut ke arduino, ini dikarenakan, sensor accelerometer tidak mau dibaca secara langsung dengan oleh lazarus yang ada di raspberry pi. Tapi lazarus raspberry pi dapat mendeteksi inputan dari GPIO. Oleh karena itu kita lewatkan data yang akan di baca di lazarus raspberry dengan terlebih dahulu mengubah bentuk data yang dari asalnya float dilewatkan ke arduino diubah ke biner, sehingga raspberry dapat membaca data dari accelerometer tersebut. Gambar pin input output arduino dan raspberry:



Gambar 3.2 GPIO Raspberry dan Arduino

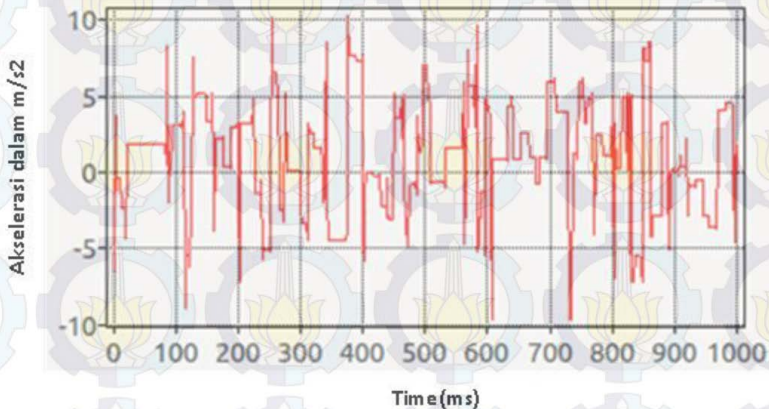
Cara yang digunakan adalah menghubungkan pin 2 sampai 13 di arduino dengan GPIO 0 sampai 12 yang tersedia di raspberry. Maka pin pin tersebut memuat nilai biner yang terbaca dari sensor accelerometer. Sehingga nilai yang terbaca tersebut dapat terbaca di lazarus raspberry. Di lazarus raspberry harus menggunakan library rpi_hal-Hardware Abstraction Library (GPIO, I2C, and SPI) function and procedures.

Library ini diciptakan oleh Stefan Fischer, library ini memuat prosedur dan fungsi untuk mengakses rpi HW I2C, SPI, dan GPIO. Fungsi dan procedure untuk GPIO yang ada di library ini:

```
procedure gpio_set_pin (pin:longword;highlevel:boolean); { Set RPi  
GPIO pin to high or low level; Speed @ 700MHz -> 0.65MHz }  
function gpio_get_PIN (pin:longword):boolean; { Get RPi GPIO pin  
Level is true when Pin level is '1'; false when '0'; Speed @ 700MHz ->  
1.17MHz }  
procedure gpio_set_input (pin:longword); { Set RPi GPIO pin to input  
direction }  
procedure gpio_set_output(pin:longword); { Set RPi GPIO pin to output  
direction }  
procedure gpio_set_alt (pin,altfunc:longword); { Set RPi GPIO pin to  
alternate function nr. 0..5 }  
procedure gpio_set_gppud (mask:longword); { set RPi GPIO Pull-  
up/down Register (GPPUD) with mask }
```

3.1.3. FFT (Fast Fourier Transform)

Nilai keluaran yang dibaca raspberry tersebut akan di baca dengan pemrograman lazarus. Sinyal yang di baca oleh raspberry adalah berupa sinyal dalam fungsi waktu (domain waktu). Hasil keluaran ini ditampilkan dalam bentuk grafik dibawah ini.



Gambar 3.3 Sinyal domain waktu keluaran accelerometer

Selanjutnya dengan FFT, sinyal waktu akan dikonversikan ke domain frekuensi. Persamaan dasar FFT ini adalah:

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn} \quad (3.1)$$

Dimana:

N , k , N adalah integer

$k = 0, 1, 2, 3, \dots, N-1$

$$j = \sqrt{-1}$$

$$W_N = e^{-j2\pi/N} = \cos \frac{2\pi}{N} - j \sin \frac{2\pi}{N} \quad (3.2)$$

Sehingga:

$$W_N^{kn} = e^{-j2\pi kn/N} = \cos \frac{2\pi kn}{N} - j \sin \frac{2\pi kn}{N} \quad (3.3)$$

Karena $x(n) = x_r(n) + jx_l(n)$ bisa bernilai kompleks, dan $X(k) = X_R(k) + jX_I(k)$ maka persamaan (3.1) menjadi :

$$\begin{aligned} X(k) &= \sum_{n=0}^{N-1} x(n) W_N^{kn} = \\ &= \sum_{n=0}^{N-1} \left[(x_r(n) + jx_l(n)) \left(\cos \frac{2\pi kn}{N} - j \sin \frac{2\pi kn}{N} \right) \right] = \\ &= \sum_{n=0}^{N-1} \left[\left(x_r(n) \cos \frac{2\pi kn}{N} - x_l(n) j \sin \frac{2\pi kn}{N} + \right. \right. \\ &\quad \left. \left. jx_l(n) \sin \frac{2\pi kn}{N} - j^2 x_r(n) \sin \frac{2\pi kn}{N} \right) \right] \end{aligned}$$

$$\begin{aligned} X(k) &= \sum_{n=0}^{N-1} \left[\left(x_r(n) \cos \frac{2\pi kn}{N} - x_l(n) \sin \frac{2\pi kn}{N} - \right. \right. \\ &\quad \left. \left. j(x_r(n) \sin \frac{2\pi kn}{N} - x_l(n) \cos \frac{2\pi kn}{N}) \right) \right] \end{aligned}$$

Karena $X(k) = X_R(k) + jX_I(k)$, maka:

$$X_R(k) = \sum_{n=0}^{N-1} \left[x_r(n) \cos \frac{2\pi kn}{N} - x_l(n) \sin \frac{2\pi kn}{N} \right] \text{ (real)} \quad (3.4)$$

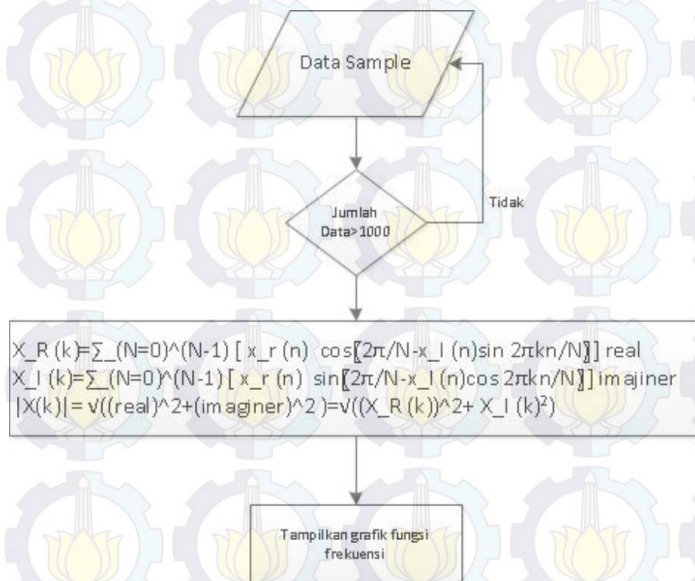
$$X_I(k) = \sum_{n=0}^{N-1} \left[x_r(n) \sin \frac{2\pi kn}{N} - x_l(n) \cos \frac{2\pi kn}{N} \right] \text{ (imajiner)} \quad (3.5)$$

Magnitude pada domain frekuensi dapat dihitung berdasarkan persamaan berikut ini:

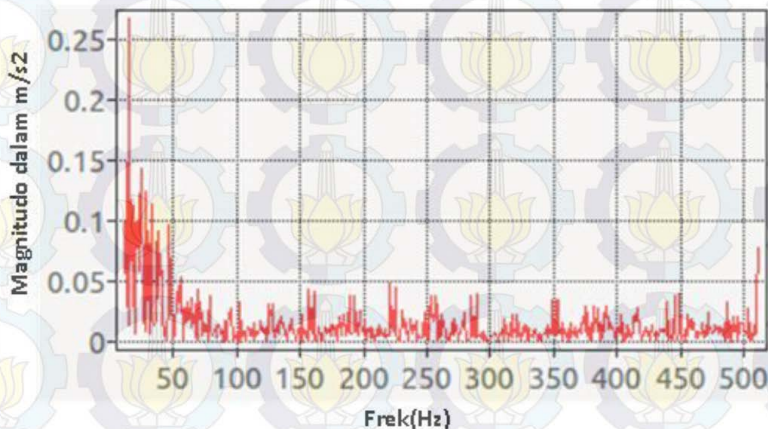
$$|X(k)| = \sqrt{(\text{real})^2 + (\text{imaginer})^2} = \sqrt{(X_R(k))^2 + X_I(k)^2} \quad (3.6)$$

Proses FFT dilakukan setiap 1000 data masukan domain waktu. Adapun flowchart untuk proses FFT adalah seperti gambar 3.4.

Setelah dilakukan proses FFT, kemudian akan ditampilkan grafik dalam fungsi frekuensi seperti gambar 3.5. Dalam domain frekuensi jika semua data frekuensi sebelah kanan sebagai pencerminan frekuensi sebelah kiri, dalam hal ini yang ditampilkan bagian kanan saja.



Gambar 3.4 Flow chart proses FFT dengan sample data vibrasi 1000 data

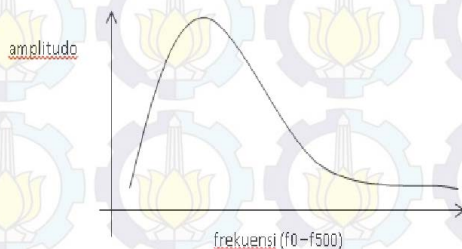


Gambar 3.5 Grafik sinyal dalam domain frekuensi

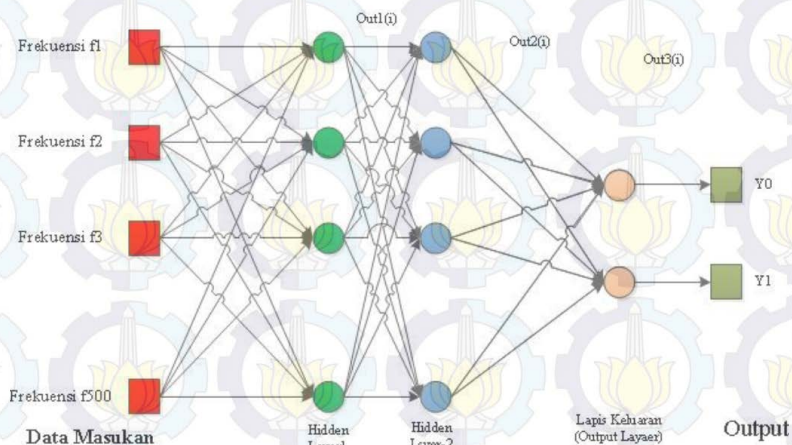
3.1.4. Neural Network

Setelah dilakukan proses FFT maka akan menghasilkan data frekuensi (f_0 sampai f_{500}). Data data frekuensi ini kemudian akan dilewatkan melalui proses neural network untuk menentukan pola vibrasi yang terjadi. Jika digambarkan secara umum grafik frekuensi akan tampak seperti gambar 3.6.

Dari grafik 3.6 tampak bahwa ada nilai frekuensi (f_0 sampai f_{500}). Nilai ini kemudian akan menjadi data masukan pada neural network. Adapun arsitektur neural network pada penelitian ini untuk pembacaan online adalah seperti gambar 3.7.



Gambar 3.6 Grafik domain frekuensi



Gambar 3.7 Arsitektur neural network dalam penelitian ini

Untuk nilai keluaran neural network (target) akan mengacu kepada table 3.1.

Tabel 3.1 Tabel hubungan pola vibrasi dengan NN

Pola Vibrasi	Target	
	Y0	Y1
Looseness (Longgar/Kendur)	0	0
Misalignment (Ketidaklurusan)	0	1
Unbalance (Ketidakseimbangan)	1	0
Unfault (Normal)	1	1

Persamaan yang digunakan dalam neural network ini pada setiap layer adalah sebagai berikut.

Persamaan umum:

$$Output = \frac{1}{1 + e^{-y}} \quad (3.7)$$

$$y = \left(\sum_{i=1}^N w(i) * x(i) \right) \quad (3.8)$$

Untuk hidden layer 1:

$$Output1(k) = \frac{1}{1 + e^{-y1(k)}} \quad (3.9)$$

$$y1(k) = \left(\sum_{i=1}^{500} w(k,i) * f(i) \right) \quad (3.10)$$

Dimana:

k = jumlah neurom hidden layer 1

f(i) = frekuensi masukan ke-i

w1(k,i) = bobot hidden layer 1

Untuk hidden layer 2:

$$Output2(j) = \frac{1}{1 + e^{-y2(j)}} \quad (3.11)$$

$$y2(j) = \left(\sum_{i=1}^k w2(i,j) * output1(i) \right) \quad (3.12)$$

Dimana:

j = jumlah neurom hidden layer 2

w2(I,j) = bobot hidden layer 2

Untuk hidden layer 3:

$$Output3(h) = \frac{1}{1 + e^{-y3(h)}} \quad (3.13)$$

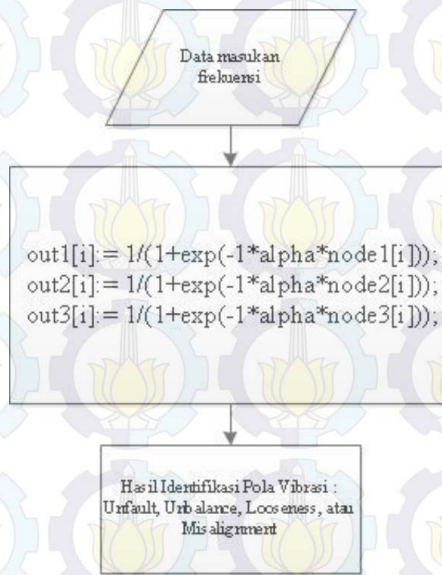
$$y3(h) = \left(\sum_{i=1}^j w3(i,h) * output2(i) \right) \quad (3.14)$$

Dimana:

h = jumlah neurom output layer

w3(i,h) = bobot output layer

Untuk nilai-niali bobot yang digunakan adalah hasil pembelajaran yang dilakukan dengan lazarus yang ada di komputer. Nilai bobot yang nantinya digunakan merupakan hasil dari nilai training dengan menggunakan backpropogation dimana nilai bobot tersebut setelah mendapatkan hasil yang diinginkan dimana hasil yang diinginkan mempunyai nilai error 0,001. Berikut adalah blok diagram proses neural network pada pengukuran online. Yang dimaksud pengukuran secara online disini adalah pengukuran secara langsung dimana ketika sensor menangkap vibrasi kemudian di teruskan ke lazarus sehingga di prose untuk mengetahui sensor sedang dalam kondisi apa, apakah unfault, unbalance, misalignment ataupun looseness.



Gambar 3.8 Flow chart proses neural network untuk identifikasi pola vibrasi

3.2. Sistem Pembelajaran Neural Network

Proses pembelajaran neural network untuk mendapatkan nilai nilai bobot yang tepat dilakukan secara terpisah menggunakan Lazarus. Langkah langkah yang dilakukan adalah sebagai berikut:

1. Melakukan pengambilan data-data untuk ke-4 kondisi (unfault, unbalance, looseness, misalignment).
2. Pembelajaran dengan lazarus menggunakan data data frekuensi yang sudah didapat.

3.2.1. Pengambilan data frekuensi untuk ke-4 pola vibrasi

Pada tahap ini dilakukan pengambilan data untuk setiap pola vibrasi. Satu paket data frekuensi berisi 500 data (f0-f500). Untuk masing-masing pola vibrasi diambil paket data sebanyak mungkin agar

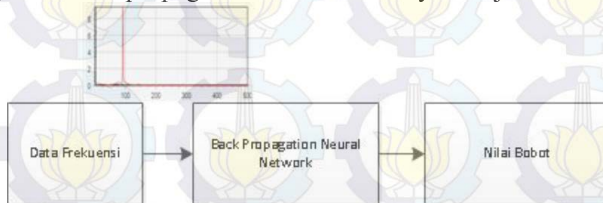
hasil proses pembelajaran dapat mendeteksi pola vibrasi dengan tingkat ketelitian yang lebih tinggi (mendekati 100%).

Cara cara untuk mendapatkan data sesuai dengan pola vibrasi akan dijelaskan pada bagian 3.3. Hasil dari pemngambilan data ini kemudian akan dimasukkan dalam pembelajaran di lazarus.

3.2.2. Pembelajaran dengan Lazarus

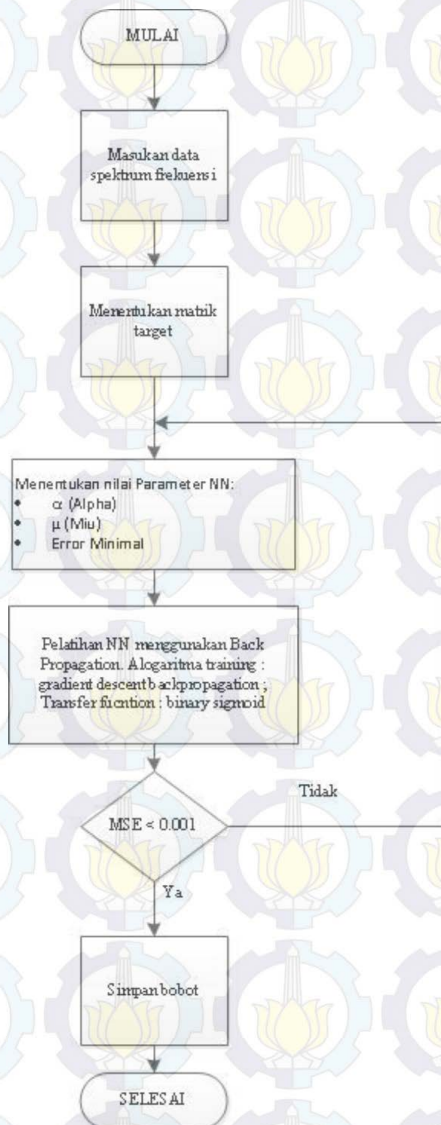
Gambar dibawah alah blok diagram untuk proses pembelajaran neural network.

Dari gambar tersebut dapat dilihat bahwa data-data frekuensi yang akan dijadikan sebagai acuan pembelajaran simasukan dalam pemrograman backpropagation NN dan akhirnya menjadi bobot.



Gambar 3.9 Blok diagram proses pembelajaran NN dengan Lazarus

Adapun flow chart pembelajaran adalah seperti pada gambar 3.10.



Gambar 3.10 Flow chart proses pembelajaran NN dengan Lazarus

Dari proses pembelajaran ini diperoleh nilai nilai bobot pada masing masing neuron. Berikut ini adalah langkah langkah dalam melakukan proses pembelajaran dengan lazarus:

1. Menentukan data yang akan detraining, dalam penelitian ini berupa data-data frekuensi. Data-data training akan meliputi untuk ke-4 pola vibrasi yaitu unfault, unbalance, looseness, dan misalignment. Dalam penelitian ini untuk setiap pola vibrasi akan diambil minimal 5 (lima) data training.
2. Menentukan nilai matrik target untuk masing masing pola vibrasi.
3. Menentukan nilai α (alpha), μ (miu) dan error minimal yang diinginkan.
4. Menentukan jenis transfer fuction untuk masing masing hidden layer dan layer output. Dalam penelitian ini akan menggunakan transfer fuction binary sigmoid $f(Vq) = 1 / (1 + \exp (-n))$
5. Menentukan jenis training. Dalam penelitian ini akan menggunakan suatu fungsi pembelajaran dengan metode backpropagation. Pembelajaran ini dimulai ketika output yang dihasilkan oleh feedforward tidak sesuai dengan yang output harapkan. Perbandingan output ini didapatkan dengan cara menghitung error pada layer output menggunakan persamaan (3.15) dan pada semua layer dengan persamaan (3.16). Setelah itu bobot nilai pada jaringan saraf buatan akan diperbarui dengan konstanta learning rate (μ) berdasarkan persamaan (3.17).

$$\delta_j^{(\ell)} = (d_{qh} - x_{out,j}^{(\ell)})g(v_j^{(\ell)}) \quad (3.15)$$

$$\delta_j^{(\ell)} = \left(\sum_{h=1}^{\ell+1} \delta_h^{(s+1)} w_{hj}^{(\ell+1)} \right) g(v_j^{(\ell)}) \quad (3.16)$$

$$w_{ji}^{(\ell)}(k+1) = w_{ji}^{(\ell)}(k) + \mu \delta_j^{(\ell)} x_{out,j}^{(\ell-1)} \quad (3.17)$$

Setelah mengambil nilai bobot pada neural network, selanjutnya data tersebut disimpan dengan perintah save di lazarus komputer.

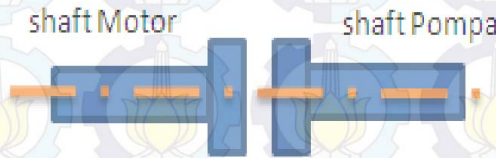
Kemudian nilai-nilai bobot ini digunakan dalam identifikasi pola vibrasi secara online.

3.3. Metode Pengambilan Data

Untuk kondisi kondisi base motor dan pompa pada saat pengambilan data pada penelitian ini saya sesuaikan dengan kondisi base motor dan pompa yang telah dilakukan oleh Moh. Ishak dalam jurnal prosedingnya yang berjudul “Analisis Proteksi Vibrasi Pada Pompa Sentrifugal Dengan Menggunakan Fast Fourier Transform Dan Neural Network” yang diterbitkan pada tahun 2009.

3.3.1. Model kondisi Unfault (Kondisi normal)

Kondisi Unfault atau kondisi normal didefinisikan sebagai kondisi normal operasi motor dan motor dapat beroperasi secara kontinyu. Salah satu ciri kondisi ini yaitu kopling pompa dapat dengan mudah diputar secara manual dengan tangan atau bisa juga dengan melepas beban. Gambar dibawah ini merupakan gambar hubungan kopling motor dan pompa pada saat kondisi unfault.

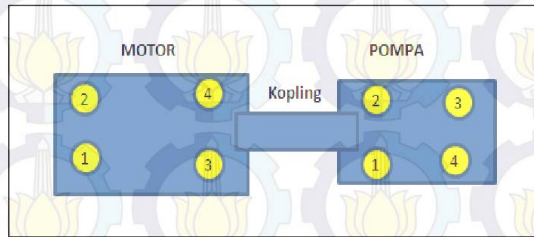


Gambar 3.11 Hubungan Kopling pompa dan motor saat Unfault

3.3.2. Model kondisi Unbalance (Ketidakseimbangan)

Kondisi unbalance yaitu kondisi dimana terjadi ketidakseimbangan pada motor ketika berputar.

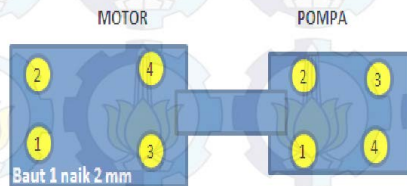
Berikut ini adalah gambar posisi-posisi baut pada base pompa dan motor pada prototipe penelitian ini.



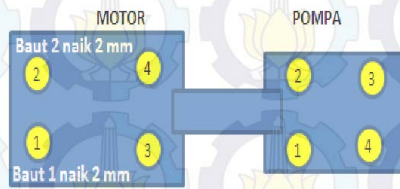
Gambar 3.12 Posisi baut pada prototype base pompa dan motor

Pada gambar diatas, angka angka 1-4 menunjukan posisi baut pada base motor dan pompa dimana pompa dan motor masing masing mempunyai 4 baut. Motor akan dikondisikan *unbalance* dengan cara menaikkan satu atau dua titik pada base motor. Hal ini dilakukan dengan cara pengaturan putaran baut.

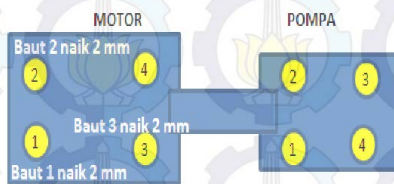
- Kondisi 1 : Base motor dinaikan pada baut no.1 sebanyak putaran baut (2mm)
- Kondisi 2 : Base motor dinaikan pada baut no.1 sebesar a putaran baut (2mm) dan pada baut no.2 sebanyak 1 putaran baut (2mm)
- Kondisi 3 : Base motor dinaikan pada baut no.1 sebesar 1 putaran baut (2mm) dan pada baut no.2 sebesar 1 putaran baut (2mm) dan pada no.3 sebesar 1 putaran baut (2mm)
- Kondisi 4 : Base motor dinaikan pada baut no.1 sebesar 1 putaran baut (2mm), pada baut no.2 sebesar 1 putaran baut (2mm), pada baut no.3 sebesar 1 putaran baut (2mm) dan pada baut no.4 sebesar 0.5 putaran baut (1mm)



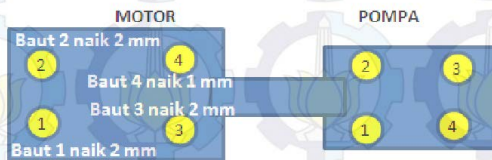
Gambar 3.13 Base motor dan pompa saat Unbalance kondisi 1



Gambar 3.14 Base motor dan pompa saat Unbalance kondisi 2



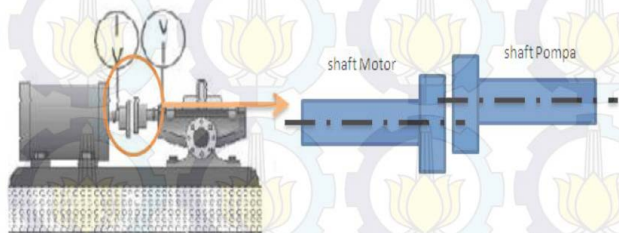
Gambar 3.15 Base motor dan pompa saat Unbalance kondisi 3



Gambar 3.16 Base motor dan pompa saat Unbalance kondisi 4

3.3.3. Model kondisi Misalignment (Tidak Selurus)

Kondisi ini didapat dengan cara menurunkan base motor dari kondisi normal. Kondisi ini ditandai dengan kondisi kopling sangat sulit untuk diputar, ini disebabkan karena kondisi hubungan kopling motor dan pompa tidak selaras. Motor pun untuk memutar kopling pompa melakukan usaha lebih, sehingga jika kondisi ini terus menerus terjadi dapat mengakibatkan kerusakan pada motor dan pompa dan kopling dikarenakan motor harus mengeluarkan tenaga lebih sehingga kecepatan motornya pun lama kelamaan menurun.



Gambar 3.17 Model kondisi Misalignment (Tidak Selurus)

3.3.4. Model kondisi Looseness (Kondisi base kendur)

Dalam penelitian ini, berikut kondisi untuk mendapatkan looseness.

- Kondisi 1: Semua baut base motor dan semua baut base pompa dalam keasaan kendur/longgar.
- Kondisi 2: Semua baut base motor dalam keadaan kendur/longgar, sementara semua baut base pompa dipasang kuat.

BAB 4

HASIL DAN ANALISA DATA

4.1. Pola-pola Spektrum Vibrasi pada Prototipe Motor

Induksi

Prototipe pada penelitian ini menggunakan motor dan pompa yang memiliki kecepatan putaran sebesar 2556 rpm (putaran per menit). Sehingga frekuensi putaran motor adalah sebagai berikut:

frekuensi putaran poros / frekuensi dsar = $\text{rpm}/(60 \text{ detik}) = 2556/60 = 42,6 \text{ Hz}$.

Sementara itu frekuensi-frekuensi harmoniknya adalah sebagai berikut:

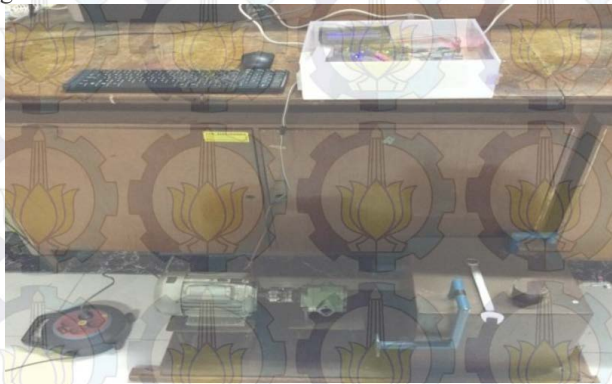
frekuensi harmonik-1 = $42,6 \times 2 = 85,4 \text{ Hz}$

frekuensi harmonik-2 = $42,6 \times 3 = 127,8 \text{ Hz}$

frekuensi harmonik-3 = $42,6 \times 4 = 170,4 \text{ Hz}$

frekuensi harmonik-4 = $42,6 \times 5 = 213 \text{ Hz}$

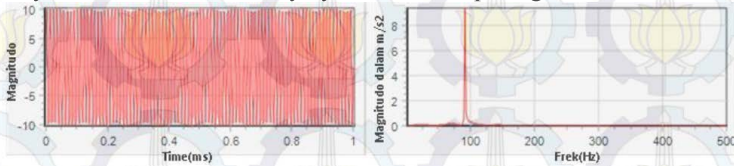
Kemudian setelah dilakukan pembuatan perangkat keras dan perangkat lunak, dilakukan perkaitan dan penyambungan prototipe seperti gambar dibawah ini.



Gambar 4.1 Foto rangkaian prototipe saat pengambilan data

Data yang dihasilkan oleh akselerometer masih berupa domain waktu, yang kemudian melalui program lazarus di raspberry data tersebut diolah menggunakan fast fourier transform menjadi domain

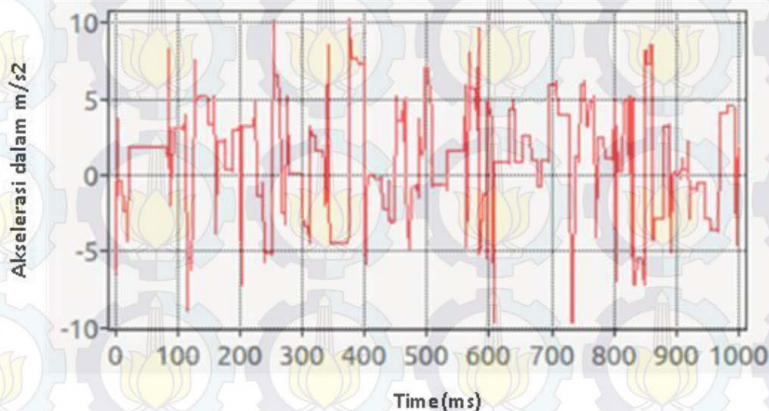
frekuensi. Untuk membuktikan bahwa pemrograman dan perhitungan FFT sudah benar atau tidak maka dilakukan pengujian dengan mensimulasikan sinyal sinusoidal. Sinyal sinusoidal yang saya berikan memiliki frekuensi 90 Hz. Hasilnya menunjukkan bahwa frekuensi yang dominan dalam hasil FFT sama dengan frekuensi yang kita inputkan di sinyal sinusoidal sebelumnya yaitu 90 Hz seperti gambar berikut ini.



Gambar 4.2 Tampilan sinyal pembuktian FFT

Pada penelitian ini, pengukuran vibrasi menggunakan 1 (satu) unit sensor akselerometer Adxl345. Sensor ini mempunyai nilai keluaran voltage, yang kemudian di konversikan menjadi g (gravitasi), dimana $1\text{ g} = 9,8\text{ m/s}^2$. Dari datashet Adxl345, disebutkan bahwa setiap keluaran tegangan 256 lsb setara dengan 1 g.

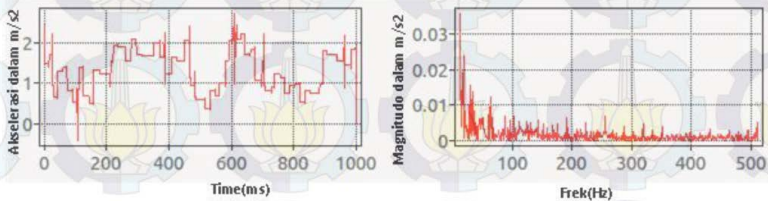
Sinyal yang ditampilkan sebagai hasil pengukuran adalah domain waktu dan domain frekuensi. Berikut ini adalah contoh domain waktu kondisi unfault.



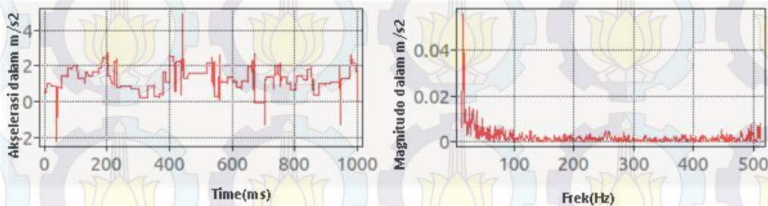
Gambar 4.3 Domain waktu kondisi unfault

4.1.1. Spektrum kondisi *Unfault* (Normal)

Seperti yang telah dijelaskan pada bab 3 di atas bahwa kondisi normal didapatkan saat motor beroperasi normal secara kontinyu. Dari sisi mekanik, dimana kondisi kopling motor dapat diputar dengan mudah secara manual dengan tangan. Pada penelitian ini, diambil beberapa data pada saat motor dinyatakan paa kondisi normal seperti gambar berikut.



Gambar 4.4 Spektrum frekuensi kondisi normal

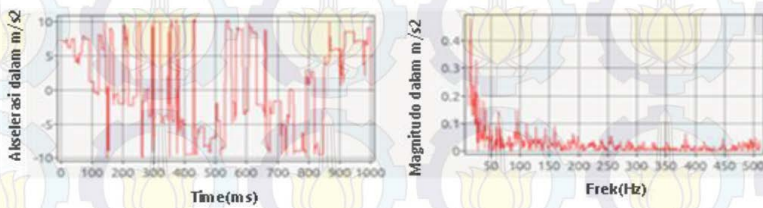


Gambar 4.5 Spektrum frekuensi kondisi normal 2

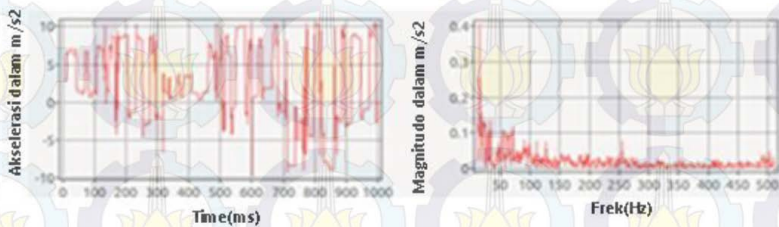
Dari spektrum kondisi normal ini, dapat dilihat bahwa amplitude tertinggi pada pada frekuensi dasar (42,6 Hz). Sementara pada frekuensi-frekuensi harmonik-1,2 dan 3 spektrum tetap muncul dengan amplitude yang lebih kecil.

4.1.2. Spektrum kondisi *Unbalance* (Ketidakseimbangan)

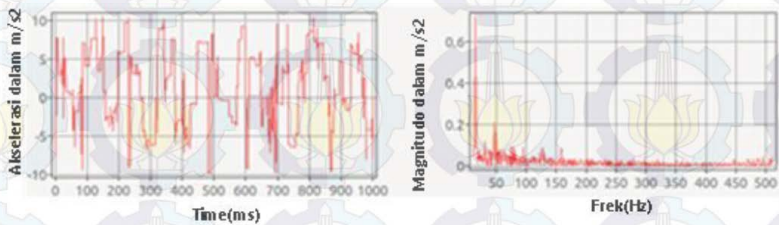
Spektrum frekuensi untuk kondisi *unbalance* terbagi menjadi 4, sesuai dengan ke empat kondisi pengambilan data yang dilakukan pada kondisi unbalance, dimana hasil spektrum vibrasi dan FFT nya pun berbeda, namun perbedaan tersebut masih masuk dalam kondisi unbalance ini. Spektrum frekuensi hasilnya adalah seperti gambar dibawah berikut:



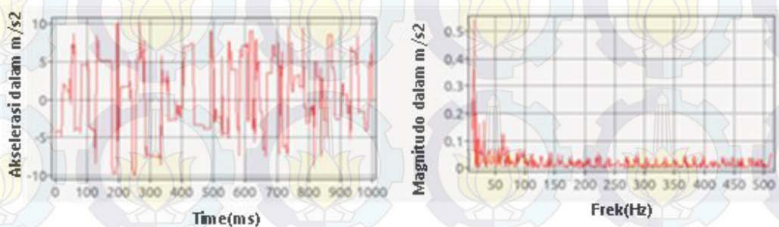
Gambar 4.6 Spektrum frekuensi unbalance kondisi 1



Gambar 4.7 Spektrum frekuensi unbalance kondisi 2



Gambar 4.8 Spektrum frekuensi unbalance kondisi 3



Gambar 4.9 Spektrum frekuensi unbalance kondisi 4

Pada gambar 4.5 dapat dilihat bahwa dengan menaikkan base motor pada kondisi 1 setinggi 2 mm maka akan menyebabkan kondisi unbalance. Hal ini disebabkan karena dengan turunnya satu titik

menyebabkan putaran poros menjadi tidak seimbang. Pada kondisi ini, amplitude pada frekuensi dasar menjadi 2 kali lebih tinggi jika dibandingkan dengan kondisi normal pada frekuensi yang sama. Sementara itu, spektrum pada frekuensi harmoniknya tetap muncul dan nilai amplitudonya tidak berubah hampir sama seperti pada kondisi normal.

Untuk kondisi 2, pola spektrum frekuensi yang akan dihasilkan masih sama dengan spektrum kondisi unbalance. Namun amplitudonya pada frekuensi dasar menurun dan lebih kecil jika dibandingkan pada kondisi 1. Hal ini disebabkan oleh posisi non drive-end motor (kiri ke kanan) mempunyai level lebih tinggi yang sama. Dan hanya berbeda dengan kondisi sisi drive-end motor.

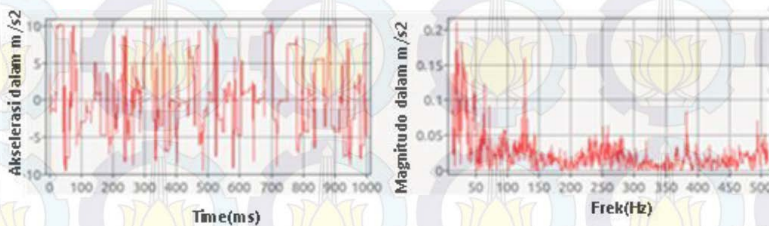
Untuk kondisi 3, amplitude pada frekuensi dasar menjadi lebih tinggi kembali mencapai lebih dari dua kali tinggi amplitude jika dibandingkan dengan kondisi normal pada frekuensi yang sama. Hal ini dikarenakan hanya satu titik yang tidak level seperti kondisi 1 sehingga menyebabkan ketidakseimbangan yang tinggi.

Pada spektrum kondisi 4, amplitude yang dihasilkan pada frekuensi dasar sama seperti dengan kondisi 3. Hal ini dikarenakan titik no.4 masih belum sama levelnya dengan ketiga titik lainnya.

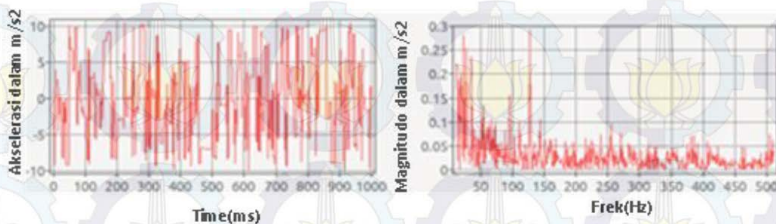
Dari analisis ke-empat spektrum frekuensi untuk kondisi unbalance, dapat disimpulkan bahwa amplitude pada frekuensi dasar menjadi hampir 3 kali besarnya jika dibandingkan dengan kondisi normal. Sementara amplitudonya pada frekuensi harmonik tetap dan hampir sama dengan kondisi normal.

4.1.3. Spektrum kondisi *Misalignment* (Tidak selurus)

Hasil spektrum frekuensi kondisi misalignment terlihat seperti gambar 4.9 dimana dapat dilihat bahwa pada saat misalignment spektrum pada frekuensi harmonik ke 2 mempunyai besar amplitude 2 kali dari amplitude pada kondisi normal. Sementara amplitude pada frekuensi lainnya tetap muncul dan besarnya sama dengan frekuensi dasar.



Gambar 4.10 Spektrum frekuensi misalignment 1



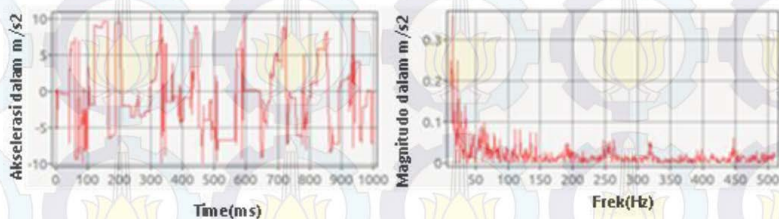
Gambar 4.11 Spektrum frekuensi misalignment 2

Hal ini sesuai dengan penelitian yang telah dilakukan oleh Giovanni Betta (IEEE, 2002) yang menyatakan bahwa kondisi misalignment akan menyebabkan amplitudo pada frekuensi harmonic akan dominan. Pada penelitian ini frekuensi harmonic ke 2 yang muncul dominan, kadang lebih tinggi dari frekuensi dasar kadang juga tidak.

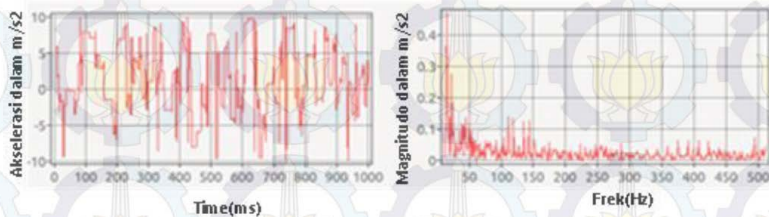
4.1.4. Spektrum kondisi *Looseness* (Kendur/longgar)

Dari gambar dibawah ini dapat dilihat bahwa pada kondisi looseness ini, frekuensi dasar mempunyai amplitude lebih tinggi jika dibandingkan dengan kondisi normal. Hal ini disebabkan karena dengan dikendurkannya semua baut pada base montor menyebabkan ketidakseimbangan menjadi lebih tinggi.

Disamping itu, dari gambar dibawah ini juga dapat dilihat bahwa muncul frekuensi harmonic lainnya pada frekuensi 112 Hz dan 144 Hz. Frekuensi frekuensi ini muncul sebagai akibat kendurnya baut baut pada base motor dan pompa.



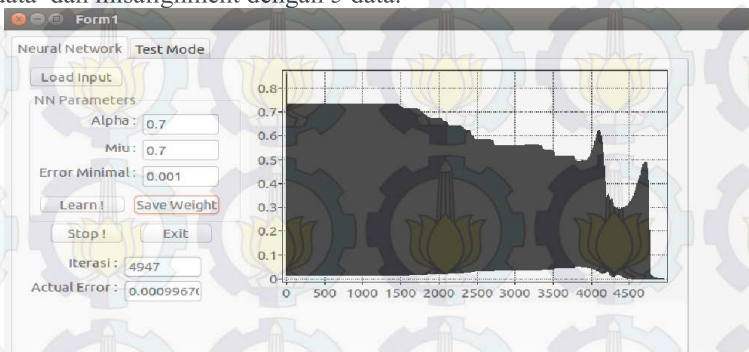
Gambar 4.12 Spektrum frekuensi looseness kondisi 1



Gambar 4.13 Spektrum frekuensi looseness kondisi 2

4.2. Klasifikasi menggunakan neural network

Untuk proses pembelajaran dilakukan dengan menggunakan lazarus di komputer. Dalam proses ini, untuk kondisi normal menggunakan 5 data, unbalance dengan 5 data, looseness dengan 10 data dan misalignment dengan 5 data.



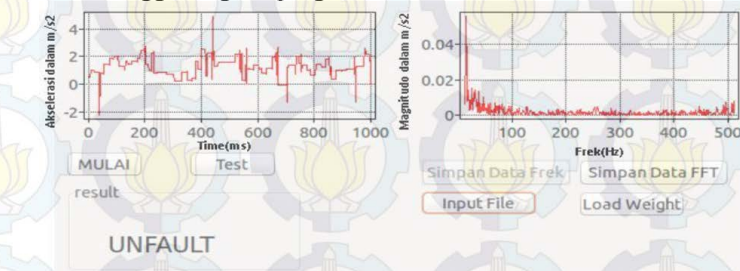
Gambar 4.14 Proses pembelajaran NN di laptop

Dari gambar diatas terlihat bahwa dengan memasukan neural network parameter α dan μ sebesar 0,7 dan menginputkan error minimal yang diinginkan adalah sebesar 0,001. Setelah proses training atau pembelajaran didapatkan bobot yang diinginkan dengan error yang diinginkan setelah melakukan 4947 literasi.

Berikut ini adalah hasil pengujian neural network dengan data yang sudah tersimpan sebelumnya.

Gambar 4.15 Hasil neural network kondisi unfault

Terlihat dalam program Neural Network diatas pada saat kita menginputkan data pada kondisi unfault mempunyai error sekitar 1,5023234 namun program neural network masih mampu mengenalinya sebagai kondisi unfault. Berikut ini adalah tampilan program yang setelah menggabungkan program neural network.



Gambar 4.16 Spektrum vibrasi kondisi unfault

Form1

Neural Network Test Mode

Load Weight

NN's Param

Alpha: 0.7

Miu: 0.7

Error Min: 0.001

Legenda

00 = Looseness
01 = Misalignment
10 = Unbalance
11 = Unfault

NN's Input

Load Input

input[1] = 0.00648
input[2] = 0.0384
input[3] = 0.0378
input[4] = 0.00631
input[5] = 0.00883
input[6] = 0.00198
input[7] = 0.00544
input[8] = 0.0133
input[9] = 0.048
input[10] = 0.0152
input[11] = 0.032

NN's Out

Out[0] = 0.999999924027721
Out[1] = 0.0017363477809200
Error = 1.50745181103888E-6

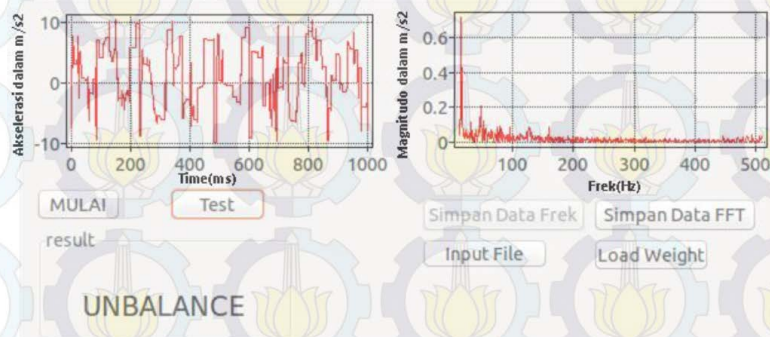
Test!

result

UNBALANCE

Gambar 4.17 Hasil neural network kondisi unbalance

Terlihat dalam program Neural Network diatas pada saat menginputkan data pada kondisi unfault mempunyai error sekitar 1,5074 namun program neural network masih mampu mengenalinya sebagai kondisi unbalance. Berikut ini adalah tampilan program yang setelah menggabungkan program neural network.



Gambar 4.18 Spektrum vibrasi kondisi unbalance

Form1

Neural Network Test Mode

Load Weight

NN's Param

Alpha : 0.7

Miu: 0.7

Error Min: 0.001

Legenda

00 = Looseness
01 = Misalignment
10 = Unbalance
11 = Unfault

NN's Input

Load Input

input[1] = 0.0311
input[2] = 0.00219
input[3] = 0.0367
input[4] = 0.0174
input[5] = 0.0066
input[6] = 0.00704
input[7] = 0.0304
input[8] = 0.0147
input[9] = 0.0272
input[10] = 0.00923
input[11] = 0.0119

NN's Out

Out[0] = 0.064477985787467
Out[1] = 0.953710200928258
Error = 0.0031500780746555

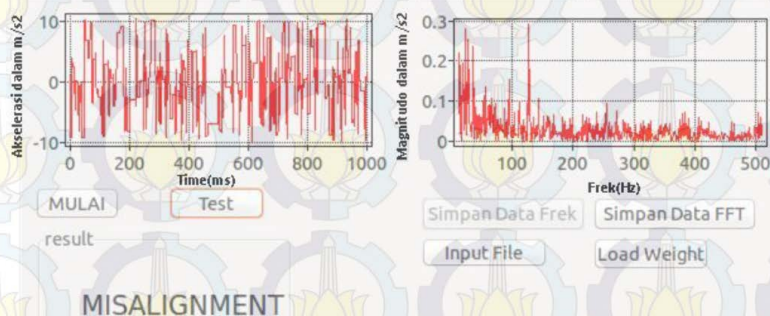
Test !

result

MISALIGNMENT

Gambar 4.19 Hasil neural network kondisi misalignment

Terlihat dalam program Neural Network diatas pada saat menginputkan data pada kondisi unfault mempunyai error sekitar 0,00315 namun program neural network masih mampu mengenalinya sebagai kondisi misalignment. Berikut ini adalah tampilan program yang setelah menggabungkan program neural network.



Gambar 4.20 Spektrum vibrasi kondisi misalignment

Form1

Neural Network Test Mode

Load Weight

NN's Param

Alpha: 0.7

Miu: 0.7

Error Min: 0.001

Legenda

00 = Looseness
01 = Misalignment
10 = Unbalance
11 = Unfault

NN's Input

Load Input

input[1] = 0.00625
input[2] = 0.0305
input[3] = 0.00898
input[4] = 0.00321
input[5] = 0.0123
input[6] = 0.0138
input[7] = 0.00381
input[8] = 0.00168
input[9] = 0.0225
input[10] = 0.00263
input[11] = 0.0171

NN's Out

Out[0] = 0.00015366054237624
Out[1] = 0.00845874027540298
Error = 0.00003578694930450

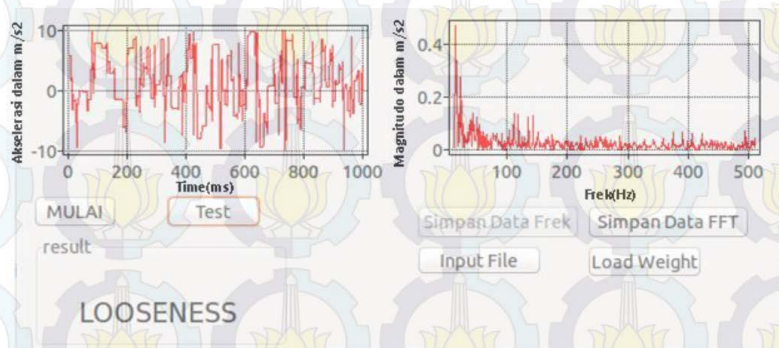
Test!

result

LOOSENESS

Gambar 4.21 Hasil neural network kondisi looseness

Terlihat dalam program Neural Network diatas pada saat menginputkan data pada kondisi unfault mempunyai error sekitar 0,00003 namun program neural network masih mampu mengenalinya sebagai kondisi looseness. Berikut ini adalah tampilan program yang setelah menggabungkan program neural network.



Gambar 4.22 Spektrum vibrasi kondisi looseness

Tabel 4.1 Hasil Pengujian NN

No.	Pola yang muncul	Hasil Neural Network	Hasil
1	Unfault	Unfault	Tepat
2	Unfault	Unfault	Tepat
3	Unbalance	Unbalance	Tepat
4	Looseness	Misalignment	Tidak Tepat
5	Misalignment	Misalignment	Tepat
6	Looseness	Looseness	Tepat
7	Unfault	Unfault	Tepat
8	Misalignment	Looseness	Tidak tepat
9	Unfault	Unfault	Tepat
10	Unbalance	Unbalance	Tepat

Dari tabel 4.1, dapat dilihat bahwa tingkat keberhasilan neural network adalah 80 %.

DAFTAR PUSTAKA

- [1] Ahmed, R. Supangat, J. Grieger, N. Ertugrul and W. L. Soong, (2004), "A Baseline Study for On-Line Condition Monitoring of Induction Machines", Australasian Universities Power Engineering Conference (AUPEC 2004).
- [2] Birajdar R, Pati R, Khanzode K, Kirloskar Brothers Ltd., (2009), "*Vibration And Noise in Centrifugal Pumps - Sources And Diagnosis Methods*", 3rd International Conference on Integrity, Reliability and Failure, Portugal, 20-24 July 2009..
- [3] Brian T. Holm-Hansen, Robert X. Gao, (2000), "Vibration Analysis of a Sensor Integrated Ball Bearing", Vol. 122, Hal 384 ~ 392, Transactions of the ASME.
- [4] Giovanni Betta, Student Member, IEEE, Consolatina Liguori, Alfredo Paolillo, and Antonio Pietrosanto, (2002), "A Dsp-Based Fft-Analyzer For The Fault Diagnosis Of Rotating Machine Based On Vibration Analysis", IEEE Transactions on Instrumentation and Measurement, vol. 51, no. 6.
- [5] H. Shreve Dennis, (1995), "Signal Processing For Effective Vibration Analysis", IRD Mechanalysis, Ins Columbus, Ohio.
- [6] Reimche W, Südmersen U, Pietsch Or, Scheer C, Bach F, (2003), "*Basics Of Vibration Monitoring For Fault Detection And Process Control*". University of Hannover, Institute of Material Science, Department of NDT.
- [7] Ruuska, Mikko, H. Andersson, Paul, Dubrovnik, (2003), "*Spindle Bearing Monitoring Using Acoustic Emission*", Proceedings, XVII IMEKO World Congress, 2164 ~ 2167.
- [8] Stephan J. Chapman. *Electric Machinery Fundamentals* (4th ed), New York: McGraw Hill, Higher Education, 2005.
- [9] Moh. Ishak, Analisis Proteksi Vibrasi Pada Pompa Sentrifugal Dengan Menggunakan Fast Fourier Transform Dan Neural Network, 2009.



Halaman ini sengaja dikosongkan

BAB 5

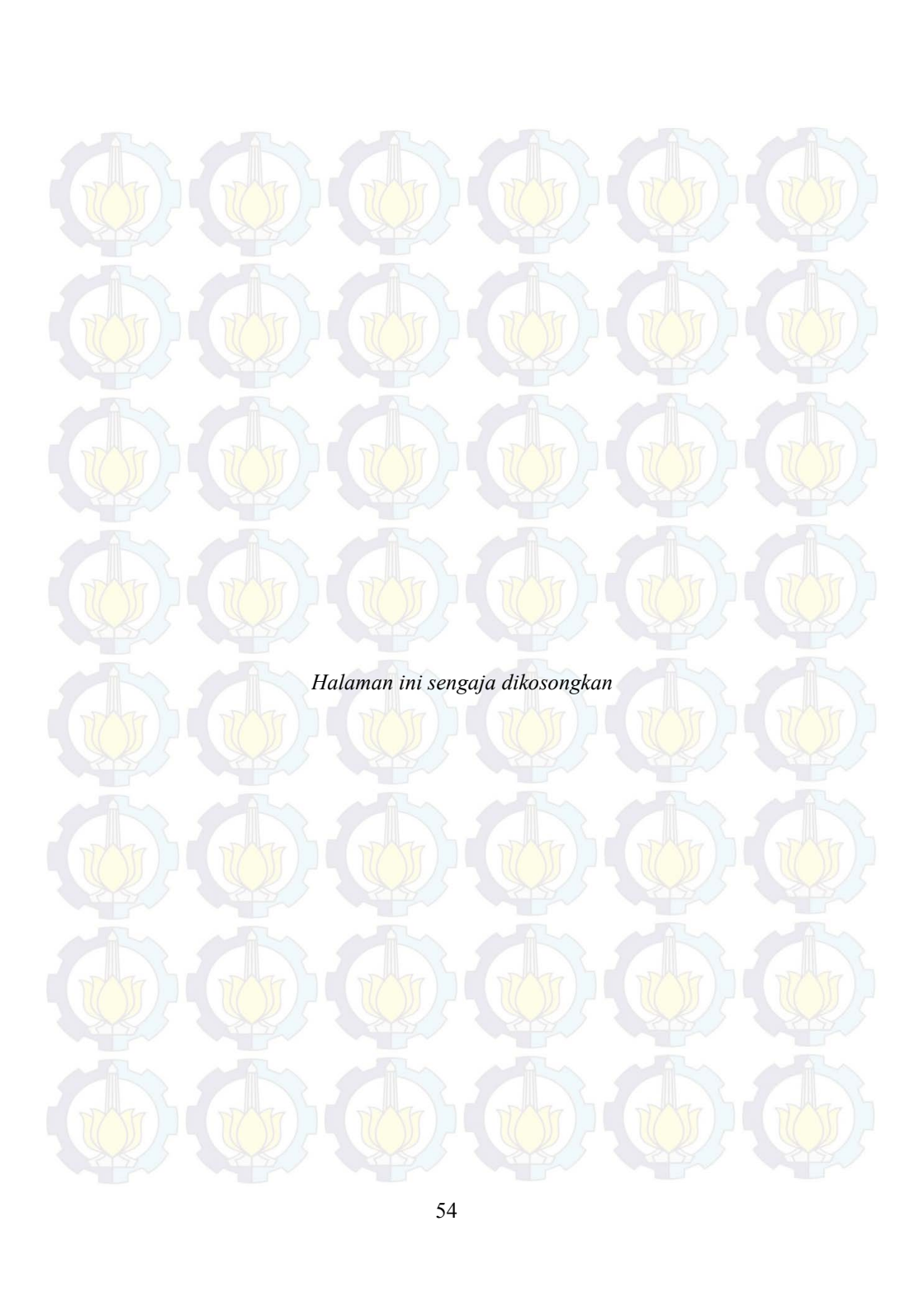
KESIMPULAN DAN SARAN

5.1. Kesimpulan

Kondisi unjuk kerja motor dapat diamati dengan menggunakan spektrum vibrasi. Kondisi unjuk kerja motor dapat diamati dengan menggunakan spektrum vibrasi. Pada kondisi normal dengan beban dan tanpa beban terjadi perbedaan tinggi amplitudo. Pada kondisi unbalance, spektrum frekuensi dasar menjadi 3 kali lebih besar dari kondisi normal. Pada kondisi misalignment, spektrum frekuensi harmonik ke-2 akan lebih dominan. Pada kondisi looseness, spektrum frekuensi dasar akan naik menjadi 2 kali lebih tinggi dari kondisi normal. Tingkat Keberhasilan neural network mencapai 80%. Dengan menggunakan sistem online spektrum vubrasi berbasis FFT dan neural network pada motor akan dapat mengetahui lebih awal kondisi kondisi alignment, unbalance, dan looseness.

5.2. Saran

Proses alignment antara pompa dan motor pada kondisi awal harus lebih presisi, karena menentukan kondisi unfault motor. Proses monitoring akan lebih mudah dan efisien jika menggunakan monitor touchscreen. Penggunaan sensor accelerometer yang standart industri lebih disarankan agar data hasil vibrasinya lebih baik. Pembuatan peletakan sensor yang kuat dan menempel dengan motor akan menghasilkan pengambilan data yang lebih baik.



Halaman ini sengaja dikosongkan

LAMPIRAN

Listing program di arduino:

```
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_ADXL345_U.h>

/* Assign a unique ID to this sensor at the same time */
Adafruit_ADXL345_Unified accel =
Adafruit_ADXL345_Unified(12345);
void outBit(float Datin1)// merubah float jadi biner
{
  //digitalWrite(2, Datin1%2);
  Datin1 = (Datin1+10)*50;
  int Datin = Datin1;
  Serial.print(Datin); Serial.print(" ");
  for(int i=2;i<12;i++) //Datin>1
  {
    Serial.print(Datin%2);
    digitalWrite(i, Datin%2); // yang dikirim ke raspberry
    Datin = Datin/2;
  }
  Serial.println("");
  return;
}

void setup(void)
{ // initialize digital pin 13 as an output.
  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);
  pinMode(8, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(10, OUTPUT);
  pinMode(11, OUTPUT);
  Serial.begin(9600);
```

```

/* Initialise the sensor */
if(!accel.begin())
{
    /* There was a problem detecting the ADXL345 ... check your
connections */
    Serial.println("Ooops, no ADXL345 detected ... Check your
wiring!");
    while(1);
}
void loop(void)
{
    /* Get a new sensor event */
    sensors_event_t event;
    accel.getEvent(&event);
    outBit(event.acceleration.y);
    //delay(8);
}

```

Listing program di lazarus raspberry pi:

```

unit Unit1;

{$mode objfpc} {$SH+}

interface


uses
    Classes, SysUtils, FileUtil, Forms, Controls, Graphics, Dialogs,
    ComCtrls,
    StdCtrls, ExtCtrls, TAGraph, TASeries, rpi_hal, unix, BaseUnix;

type

    { TForm1 }

    TForm1 = class(TForm)
        Button1: TButton;
        Button2: TButton;
        Button4: TButton;

```

```

Button5: TButton;
Button6: TButton;
ButtonInputFile: TButton;
ButtonMulai: TButton;
Chart1: TChart;
Chart1LineSeries1: TLineSeries;
Chart1LineSeries2: TLineSeries;
Chart1LineSeries3: TLineSeries;
Chart1LineSeries4: TLineSeries;
Chart2: TChart;
Chart2LineSeries1: TLineSeries;
Chart2LineSeries2: TLineSeries;
Chart2LineSeries3: TLineSeries;
Chart2LineSeries4: TLineSeries;
Edit1: TEdit;
Edit2: TEdit;
GroupBox6: TGroupBox;
Label16: TLabel;
OpenDialog1: TOpenDialog;
OpenDialog2: TOpenDialog;
OpenDialog3: TOpenDialog;
SaveDialog1: TSaveDialog;
SaveDialog2: TSaveDialog;
Timer1: TTimer;
procedure Button1Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure Button4Click(Sender: TObject);
procedure Button5Click(Sender: TObject);
procedure Button6Click(Sender: TObject);
procedure ButtonInputFileClick(Sender: TObject);
procedure ButtonMulaiClick(Sender: TObject);
procedure ButtonSettingClick(Sender: TObject);
procedure ComboBoxJumlahDataChange(Sender: TObject);
procedure ComboBoxNilaiMaksimumChange(Sender: TObject);
procedure FormClose(Sender: TObject; var CloseAction:
TCloseAction);
procedure FormCreate(Sender: TObject);
procedure PageControl1Change(Sender: TObject);

```

```

procedure ComPort1RxChar();
procedure load_weight();
procedure load_input();
procedure feedforward();
procedure backpropagation();
private
{ private declarations }
public
{ public declarations }
end;

var
Form1: TForm1;
accel_data: array[1..4,1..10000] of extended;
data_frekuensi: array[1..10000] of extended;
sst_state,sst_data_length: integer;
rx_index: byte;
rx_buff: array[1..250] of byte;
str, nilai, sumbu: string;
com_state: boolean;
cksum,temp: byte;
data_index,data_buffer: integer;
xasali,xasali: array[1..10000] of extended;
N,index,panjang_fft: integer;
x1,xi,xr,phi,power,w: array[1..10000] of extended;
filename: textfile;
indexfft: integer;
amp: array[1..4,1..10000] of extended;
nambah : integer;

z: integer;
iterasi: integer;
miu,alpha,erormin,error: extended;
temp2: Textfile;
tombol: integer;
buff: string;
erkum: array[1..25] of extended;
input: array[1..26,1..500] of extended;

```

```

node1,turunan1,out1,error1: array[1..60] of extended;
node2,turunan2,out2,error2: array[1..20] of extended;
node3,turunan3,out3,error3: array[1..2] of extended;
w1: array[1..500,1..60] of extended;
w2: array[1..60,1..20] of extended;
w3: array[1..20,1..2] of extended;

const
    desired_output: array[1..25,1..2] of integer=
((0,0),(0,0),(0,0),(0,0),(0,0),
(0,0),(0,0),(0,0),(0,0),(0,0),
(0,1),(0,1),(0,1),(0,1),(0,1),
(1,0),(1,0),(1,0),(1,0),(1,0),
(1,1),(1,1),(1,1),(1,1),(1,1));

implementation
{$SR *.lfm}

procedure TForm1.ButtonSettingClick(Sender: TObject);
begin
    //ComPort1.ShowSetupDialog;
end;

procedure TForm1.PageControl1Change(Sender: TObject);
begin
end;

// Program FFT //
procedure fft(t:extended);
label satu,dua,tiga,empat;
var
    lim1,lim2,lim3,l,r,m,k,i,j:integer;
    d,x1,x2,b1,b2,c1,c2,arg,sin1,cos1:extended;
begin
    m:= N;
    for k:= 1 to panjang_fft do

```



```

begin
  xr[k]:=xasalr[k];
  xi[k]:=xasali[k];
end;

if t=1.0 then
  d:=1.0
else
  d:=panjang_fft;

  {suffle data input}
  lim1:=panjang_fft-1;
  lim2:= round(panjang_fft/2);

  j:= 1;
  for i:= 1 to lim1 do
    begin
      if i > j-1 then
        goto dua;
      x1:=xr[j];
      x2:=xi[j];
      xr[j]:=xr[i];
      xi[j]:=xi[i];
      xr[i]:=x1;
      xi[i]:=x2;
    dua:
      l:=lim2;
    tiga:
      if l > j-1 then goto empat;
      j:=j-l;
      l:=round(l/2);
      goto tiga;
    empat:
      j:=j+l;
    end;
  }inplace transformation}
  for i:= 1 to m do
    begin

```

```

lim1:= round(exp((i-1)*ln(2)));
lim2:= round(exp((m-i)*ln(2)));
for l:= 1 to lim2 do
begin
  for r:= 1 to lim1 do
  begin
    lim3:= (r-1)+(l-1)*2*lim1+1;
    b1:=xr[lim3];
    b2:=xi[lim3];
    c1:=xr[lim3+lim1];
    c2:=xi[lim3+lim1];
    arg:=2*pi*(r-1)*lim2/panjang_fft;
    cos1:=cos(arg);
    sin1:=sin(arg);
    x1:=(c1*cos1)+(c2*sin1*t);
    x2:=(-c1*sin1*t)+(c2*cos1);
    xr[lim3]:=b1+x1;
    xi[lim3]:=b2+x2;

    xr[lim3+lim1]:=b1-x1;
    xi[lim3+lim1]:=b2-x2;
  end;
end;

end;
for k:= 1 to panjang_fft do
begin
  xr[k]:=xr[k] / d;
  xi[k]:=xi[k] / d;
end;

end;

// Biner To Integer //
function BinToInt(value: string): Integer;
var
  i, iValueSize: Integer;
begin
  Result := 0;

```

```

iValueSize := Length(value);
for i:= iValueSize downto 1 do
if Value[i] = '1' then Result := Result + (1 shl (iValueSize - i));
end;

```

```

procedure TForm1.ComPort1RxChar();
var
str: string;
sst_state,sst_data_length,chk: byte;
j,k,l,m: integer;
i: byte;
begin
data_index:=1;
Chart1LineSeries1.Clear;
while data_index < 1000 do
begin
nilai:="";
sumbu:="";

for l:=11 downto 2 do
begin
rpi_hal.gpio_set_input(l);
if rpi_hal.gpio_get_PIN(l) then nilai:= nilai +'1'
else nilai:= nilai +'0';
end;

```

```

Edit2.Text:=nilai;
accel_data[1][data_index]:=(BinToInt(nilai) / 50)-10;

```

```

Edit1.Text:=FloatToStr(BinToInt(nilai));

```



```
Chart1LineSeries1.AddXY(data_index,(accel_data[1][data_index]));
```

```
data_index:=data_index+1;
```

```
//sleep(9); //delay 1 milisecond
```

```
//Application.ProcessMessages;
```

```
if data_index=1000 then
```

```
begin
```

```
for j :=1 to 1 do
```

```
begin
```

```
for k:= 1 to data_index do //1024 do
```

```
begin
```

```
xasalr[k]:=accel_data[j][k]; //data_frekuensi[i];
```

```
xasali[k]:=0.0;
```

```
end;
```

```
N:=round(ln(data_index)/ln(2));
```

```
panjang_fft:=1 shl N ;
```

```
fft(0.0);
```

```
indexfft:=panjang_fft div 2;
```

```
for k:= indexfft+1 to panjang_fft do
```

```
begin
```

```
amp[j][k-indexfft]:= sqrt(xr[k]*xr[k]+xi[k]*xi[k]);
```

```
end;
```

```
for k:= 1 to panjang_fft div 2 do
```

```
begin
```

```
amp[j][k+indexfft]:= sqrt(xr[k]*xr[k]+xi[k]*xi[k]);
```

```
end;
```

```
Chart2LineSeries1.Clear;
```

```
for k:= 1 to round(data_index/2) do
```

```
begin
```

```
Chart2LineSeries1.AddXY(indexfft-k,amp[j][k]);
```

```
end;
```

```
Button1.Enabled:=true;
```

```
Button2.Enabled:=true;
```

```
panjang_fft:=data_index;
```

```

        end;
        // data_index:=1;
    end;

end;
for n:= 1 to 500 do
begin
    input[26,n]:= amp[1][n];
end;
end;

```

```

procedure TForm1.FormCreate(Sender: TObject);
begin
    nambah := 0;
    Button1.Enabled:=false;
    Button2.Enabled:=false;
    data_index:=1;
end;

```

```

procedure TForm1.ButtonMulaiClick(Sender: TObject);
begin
    if ButtonMulai.Caption='MULAI' then
    begin
        ButtonMulai.Caption:='BERHENTI';
        ComPort1RxChar();
    end
    else
    begin
        cksum:=0;
        cksum:=$1E+$28+$01+$00+$01+$01+$00;
        cksum:=0-cksum;
        ButtonMulai.Caption:='MULAI';
        rx_index:=1;
    end;
end;

```

```

end;

procedure TForm1.ComboBoxJumlahDataChange(Sender: TObject);
begin

end;

procedure TForm1.FormClose(Sender: TObject; var CloseAction:
TCloseAction);
begin

end;

procedure TForm1.ComboBoxNilaiMaksimumChange(Sender:
TObject);
begin

end;

procedure TForm1.ButtonInputFileClick(Sender: TObject);
var
jumlah_data,i: integer;
number: array[1..10000] of integer;
j: extended;
begin
  if OpenFileDialog1.Execute then
  begin
    AssignFile(filename,OpenDialog1.FileName);
    reset(filename);
    jumlah_data:=1;
    while not (eof(filename)) do
    begin
      readln(filename,data_fre[k[jumlah_data]]);
      inc(jumlah_data);
    end;
    jumlah_data:=jumlah_data-1;
    j:=jumlah_data/20+10;
    closefile (filename);
    Chart1LineSeries1.Clear;
  end;
end;

```



```

for i:= 1 to jumlah_data do //1024 do
begin
  j:=i/20;
  Chart1LineSeries1.AddXY(i,data_frekuensi[j]);
end;
for i:= 1 to jumlah_data do //1024 do
begin
  xasalr[i]:=data_frekuensi[j];
  xasali[i]:=0.0;
end;
N:=round((ln(jumlah_data)/ln(2)));
panjang_fft:=1 shl N ;
fft(0.0);
indexfft:=panjang_fft div 2;
for i:= indexfft+1 to panjang_fft do
begin
  amp[1][i-indexfft]:= sqrt(xr[i]*xr[i]+xi[i]*xi[i]);
end;
for i:= 1 to panjang_fft div 2 do
begin
  amp[1][i+indexfft]:= sqrt(xr[i]*xr[i]+xi[i]*xi[i]);
end;
Chart2LineSeries1.Clear;
for i:= 1 to round(jumlah_data/2) do
begin
  Chart2LineSeries1.AddXY(indexfft-i,amp[1][i]);
end;
Button2.Enabled:=true;
panjang_fft:=jumlah_data;

end;
for i:= 1 to 500 do
begin
  input[26,i]:= amp[1][i];
end;
end;

procedure TForm1.Button2Click(Sender: TObject);

```

```

var
i: integer;
begin
  if SaveDialog1.Execute then
  begin
    SaveDialog1.FileName:=SaveDialog1.FileName+'.txt';
    AssignFile(filename,SaveDialog1.FileName);
    rewrite(filename);
    for i:= 1 to round(panjang_fft/2) do
      begin
        writeln(filename,#$09,amp[1][i]);//,$$09,amp[2][i],

      end;
      closefile(filename);
    end;
  end;

procedure TForm1.Button3Click(Sender: TObject);
begin
end;

procedure TForm1.Button4Click(Sender: TObject);
var
l : integer;
begin
  nambah := nambah + 1;
  nilai:="";
  sumbu:="";

  for l:=8 downto 1 do
  begin
    rpi_hal.gpio_set_input(l);
    if rpi_hal.gpio_get_PIN(l) then nilai:= nilai +'1'
    else nilai:= nilai +'0';
  end;

```

```

Edit1.Text:=IntToStr((BinToInt(nilai) div 10)-10);
Chart1.LineSeries1.AddXY(nambah,((BinToInt(nilai) div 10)-10));
end;

procedure TForm1.Button1Click(Sender: TObject);
var
i: integer;
begin
if SaveDialog1.Execute then
begin
SaveDialog1.FileName:=SaveDialog1.FileName+'.txt';
AssignFile(filename,SaveDialog1.FileName);
rewrite(filename);
ShowMessage(IntToStr(data_index));
for i:= 1 to data_index do
begin

writeln(filename,#$09,accel_data[1][i]);//,$$09,accel_data[2][i],#$09,

end;
closefile(filename);
end;
end;

procedure tform1.load_weight;
var
i,j: integer;
begin
if OpenFileDialog1.Execute then
begin
assignfile(temp2,opendialog1.FileName);
reset(temp2);
for i:= 1 to 500 do
begin
for j:= 1 to 60 do
begin
readln(temp2,buff);
w1[i,j]:= strtofloat(buff);

```



```

end;
end;

for i:= 1 to 60 do
begin
  for j:= 1 to 20 do
  begin
    readln(temp2,buff);
    w2[i,j]:= strtocfloat(buff);
  end;
end;

for i:= 1 to 20 do
begin
  for j:= 1 to 2 do
  begin
    readln(temp2,buff);
    w3[i,j]:= strtocfloat(buff);
  end;
end;
readln(temp2,buff);
alpha:= strtocfloat(buff);

readln(temp2,buff);
miu:= strtocfloat(buff);

readln(temp2,buff);
erormin:= strtocfloat(buff);

closefile(temp2);
end;
end;

procedure TForm1.load_input;
begin
end;

```

```
procedure TForm1.Button5Click(Sender: TObject);  
begin  
    load_weight;  
end;
```

```
procedure TForm1.Button6Click(Sender: TObject);  
var  
    x,y: integer;  
    init_eror: extended;  
    pos: integer;  
begin  
    z:= 26;  
    feedforward;
```

```
    init_eror:= 1;
```

```
    for x:= 1 to 25 do  
    begin  
        erkum[x]:=0;  
        for y:=1 to 2 do  
        begin
```

```
            erkum[x]:= erkum[x]+sqr(desired_output[(x),y]-out3[y]);
```

```
        end;
```

```
        erkum[x]:= 0.5*(erkum[x]);
```

```
        if erkum[x]< init_eror then
```

```
        begin
```

```
            init_eror:=erkum[x];
```

```
            pos:= x;
```

```
        end;
```

```
    end;
```

```
    if (pos<=10) then
```

```
    begin
```

```
        label16.Caption:= 'LOOSENESS';
```

```
    end
```

```
    else if (pos>10) and (pos<= 15) then
```

```
    begin
```

```
        label16.Caption:= 'MISALIGNMENT';
```

```

end
else if (pos>15) and (pos<= 20) then
begin
    label16.Caption:= 'UNBALANCE';
end
else if (pos>20) and (pos<= 25) then
begin
    label16.Caption:= 'UNFAULT';
end
else
begin
    label16.Caption:= 'UNKNOWN';
end;
end;

procedure tform1.feedforward;
var
    i,j: integer;
begin
    for i:= 1 to 60 do
    begin
        node1[i]:= 0;
        for j:= 1 to 500 do
        begin
            node1[i]:= node1[i]+(w1[j,i]*input[z,j]);
        end;
        out1[i]:= 1/(1+exp(-1*alpha*node1[i]));
        turunan1[i]:= alpha*out1[i]*(1-out1[i]);
    end;

    for i:= 1 to 20 do
    begin
        node2[i]:= 0;
        for j:= 1 to 60 do
        begin
            node2[i]:= node2[i]+(w2[j,i]*out1[j]);
        end;
        out2[i]:= 1/(1+exp(-1*alpha*node2[i]));
        turunan2[i]:= alpha*out2[i]*(1-out2[i]);
    end;
end;

```



```

end;

for i:= 1 to 2 do
begin
    node3[i]:= 0;
    for j:= 1 to 20 do
        begin
            node3[i]:= node3[i]+(w3[j,i]*out2[j]);
        end;
        out3[i]:= 1/(1+exp(-1*alpha*node3[i]));
        turunan3[i]:= alpha*out3[i]*(1-out3[i]);
    end;
end;

```

```

procedure tform1.backpropagation;
var
    i,j: integer;
begin
    for i:= 1 to 2 do //error output (layer 3)
        begin
            eror3[i]:= (desired_output[z,i]-out3[i])*turunan3[i];
        end;
        erkum[z]:= 0;

        for i:= 1 to 2 do
            begin
                erkum[z]:= erkum[z]+sqr(desired_output[z,i]-out3[i]);
            end;

            erkum[z]:= (0.5)*erkum[z];

            for i:=1 to 20 do
                begin
                    eror2[i]:= 0;
                    for j:= 1 to 2 do
                        begin
                            eror2[i]:= eror2[i]+(eror3[j]*w3[i,j]);

```



```
end;  
error2[i]:= error2[i]*turunan2[i]; //error layer 2  
end;
```

```
for i:=1 to 60 do  
begin  
error1[i]:= 0;  
for j:= 1 to 20 do  
begin  
error1[i]:= error1[i]+(error2[j]*w2[i,j]);  
end;  
error1[i]:= error1[i]*turunan1[i]; //error layer 1  
end;  
end;
```

```
end.
```

```
end.
```



Halaman ini sengaja dikosongkan